Université de Montréal

Traduction statistique vers une langue à morphologie riche : Combinaison d'algorithmes de segmentation morphologique et de modèles statistiques de traduction automatique

par
Chiheb Trabelsi

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maitrise ès Science (M. Sc.)
en Informatique

Juillet, 2012

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé :

Traduction statistique vers une langue à morphologie riche : Combinaison d'algorithmes de segmentation morphologique et de modèles statistiques de traduction automatique

Présenté par :
Chiheb Trabelsi

Évalué par un jury composé des personnes suivantes :

Jian-Yun Nie, président-rapporteur
Philippe Langlais, directeur de recherche
Neil Stewart, membre du jury

Résumé

Les systèmes statistiques de traduction automatique ont pour tâche la traduction d'une langue source vers une langue cible. Dans la plupart des systèmes de traduction de référence, l'unité de base considérée dans l'analyse textuelle est la forme telle qu'observée dans un texte. Une telle conception permet d'obtenir une bonne performance quand il s'agit de traduire entre deux langues morphologiquement pauvres. Toutefois, ceci n'est plus vrai lorsqu'il s'agit de traduire vers une langue morphologiquement riche (ou complexe).

Le but de notre travail est de développer un système statistique de traduction automatique comme solution pour relever les défis soulevés par la complexité morphologique. Dans ce mémoire, nous examinons, dans un premier temps, un certain nombre de méthodes considérées comme des extensions aux systèmes de traduction traditionnels et nous évaluons leurs performances. Cette évaluation est faite par rapport aux systèmes à l'état de l'art (système de référence) et ceci dans des tâches de traduction anglais-inuktitut et anglais-finnois. Nous développons ensuite un nouvel algorithme de segmentation qui prend en compte les informations provenant de la paire de langues objet de la traduction. Cet algorithme de segmentation est ensuite intégré dans le modèle de traduction à base d'unités lexicales « **Phrase-Based Models** » pour former notre système de traduction à base de séquences de segments. Enfin, nous combinons le système obtenu avec des algorithmes de post-traitement pour obtenir un système de traduction complet. Les résultats des expériences réalisées dans ce mémoire montrent que le système de traduction à base de séquences de segments proposé permet d'obtenir des améliorations significatives au niveau de la qualité de la traduction en terme de le métrique d'évaluation **BLEU** (Papineni et al., 2002) et qui sert à évaluer. Plus particulièrement, notre approche de segmentation réussie à améliorer légèrement la qualité de la traduction par rapport au système de référence et une amélioration significative de la qualité de la traduction est observée par rapport aux techniques de prétraitement de base (baseline).

Mots-clés : traduction statistique, apprentissage automatique, traitement automatique de la langue, complexité morphologique, génération morphologique, segmentation.

Abstract

Statistical Machine Translation systems have been designed to translate text from a source language into a target one. In most of the benchmark translation systems, the basic unit considered in the textual analysis is the observed textual form of a word. While such a design provides good performance when it comes to translation between two morphologically poor languages, this is not the case when translating into or from a morphologically rich (or complex) language.

The purpose of our work is to develop a Statistical Machine Translation (**SMT**) system as an alternative solution to the many challenges raised by morphological complexity. Our system has the potentials to capture the morphological diversity and hence, to produce efficient translation from a morphologically poor language to a rich one. Several methods have been designed to accomplish such a task. Pre-processing and Post-processing techniques have been built-in to these methods to allow for morphological information to improve translation quality. In this thesis, we first examine several methods of extending traditional **SMT** models and assess their power of producing better output by comparing them on English-Inuktitut and English-Finnish translation tasks. In a second step we develop a new morphologically aware segmentation algorithm that takes into account information coming from both languages to segment the morphologically rich language. This is done in order to enhance the quality of alignments and consequently the translation itself. This bilingual segmentation algorithm is then incorporated into the phrase-based translation model "**PBM**" to form our segmentation-based system. Finally we combine the segmentation-based system thus obtained with post-processing algorithms to procure our complete translation system. Our experiments show that the proposed segmentation-based system slightly outperforms the baseline translation system which doesn't use any preprocessing techniques. It turns out also that our segmentation approach significantly surpasses the preprocessing baseline techniques used in this thesis.

Keywords: statistical machine translation, statistical machine learning, natural language processing, morphological complexity, morphology generation, word segmentation.

Table des matières

Liste des tableaux

Liste des figures

À papa, à maman et à Amine

Remerciements

Je souhaite adresser mes remerciements les plus sincères à toutes les personnes qui m'ont apporté de l'aide et qui ont contribué, de prés ou de loin, à la réalisation de ce mémoire.

Je tiens, tout d'abord, à exprimer mon amour et ma reconnaissance envers mes parents, Monia et Abdelwahed, ainsi qu'à mon grand frère Amine, qui m'ont apporté un soutien autant moral que matériel et sans lequel, je n'aurais jamais pu arriver à ce stade de mes études.

Je tiens, particulièrement, à remercier mon directeur de recherche, monsieur le professeur Philippe Langlais qui s'est montré, toujours, à l'écoute et très disponible tout au long de la réalisation de ce projet, ainsi que pour l'inspiration, l'aide et le temps qu'il a bien voulu me consacrer et sans lequel, ce mémoire n'aurait jamais pu prendre fin.

J'exprime aussi mes sincères remerciements à Ann Clifton et à Anoop Sarkar qui ont accepté de nous fournir leurs données finnoises qui leur ont servi à entraîner et à évaluer leurs systèmes de traduction dans le cadre du mémoire de maîtrise de Ann en 2010.

Par ailleurs, j'aimerais exprimer ma gratitude à monsieur Hafedh El Ayech qui a consacré énormément de son temps pour m'apprendre l'art de la programmation et les fondements de l'apprentissage automatique. Ma réussite dans les études secondaires et universitaires n'aurait jamais pu se réaliser sans son aide inconditionnée.

J'adresse, pareillement, une pensée particulière à monsieur Mohammed Bouaziz qui m'a enseigné les mathématiques lors de mon année terminale au lycée et qui m'a inculqué la logique mathématique qui m'a permis d'appréhender et de résoudre les problèmes philosophiques, mathématiques et algorithmiques rencontrés.

Finalement, je voudrais exprimer mes sentiments de reconnaissance et de gratitude à tous mes proches et mes amis qui m'ont toujours soutenu et encouragé tout au long de mon

aventure estudiantine et en particulier mon cousin Aymen, qui m'a été d'un grand support moral durant l’année et demie que nous avons passé ensemble à Montréal.

Chapitre 1 Introduction

La traduction statistique ou en anglais « Statistical Machine Translation (**SMT**) » est la science qui étudie la traduction automatique d'une langue vers une autre à l'aide de techniques statistiques. La langue à partir de laquelle on traduit est appelée langue source et celle vers laquelle on traduit est appelée langue cible. La tâche principale des modèles **SMT** est de trouver la traduction la plus probable pour une séquence de mots appartenant à une langue source. Plus spécifiquement, cette tâche consiste à développer des modèles probabilistes capables d'apprendre à traduire, d'une langue vers une autre. Cet apprentissage peut être fait de manière supervisée ou non supervisée, en s'appuyant sur des textes parallèles, de tailles importantes, relatifs aux langues source et cible. Les techniques **SMT** traditionnelles s'avèrent performantes quand il s'agit de traduire vers l'anglais à partir des langues ayant des structures morphologiques qui lui sont similaires. Ces techniques traditionnelles sont de deux types. Le premier type est appelé traduction à base de mots « Word-Based Translation (**WBT**) » et le second est nommé traduction à base de séquences (ou segments) de mots. « Phrase-Based Translation (**PBT**) ». Ces deux techniques seront mentionnées avec plus de détails dans le chapitre 2.

Les techniques traditionnelles présentent certaines limites. Ces limites deviennent de plus en plus concrètes et accentuées lorsqu'il s'agit d'appréhender des langues morphologiquement différentes. Pour ces techniques traditionnelles, cette sous-performance pourrait être attribuée, entre autres, à l'hypothèse qui consiste à prendre le mot comme base ou « unité atomique » de traduction et d'analyse. En effet cette hypothèse s'avère restrictive et même problématique en dehors des langues à morphologie relativement simples. En linguistique, les langues sont classées selon une échelle qui traduit le degré de complexité morphologique. La construction de cette échelle fait appel à la notion de morphèmes qui sera évoquée ultérieurement.

Le principe adopté par le premier type de techniques traditionnelles est donc basé sur la traduction de mots (**WBT**). Le mot est considéré alors comme l'unité de traduction la

plus petite. Dans certaines combinaisons de langues, les alignements sont, dans la plupart des cas, mot à mot. L'alignement entre les mots français et les mots anglais, schématisé par la figure 1.1, illustre une telle correspondance entre les mots des deux langues.

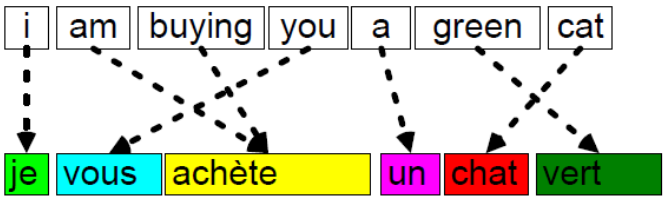

Figure 1.1 : Alignement entre une phrase anglaise et sa traduction en français (Koehn et al., 2007).

Les systèmes de traduction à base de mots peuvent s'avérer utiles dans des tâches de traduction incluant de telles langues. Cependant, un mot dans une langue peut être traduit par plusieurs mots dans la langue cible ou vice versa. En finnois par exemple le mot ''*järjestelmässämme*'' est traduit en anglais par ''*in our system*''. Dans la langue inuktitute qui est parlée dans le territoire du Nunavut au Canada, l'alignement des mots dans une paire de phrases n'est pas établi mot à mot. Cela est illustré dans l'exemple mentionné par (Johnson et Martin, 2003), où le mot en inuktitut ''*qaisaaliniaqquunngikkaluaqpuq*'' est traduit par 8 mots anglais ''*Actually he will probably not come early today*''.

Le deuxième type de technique traditionnelle, qui est l'approche (**PBT**) et qui a été proposée par (Koehn, Och, et Marcu, 2003), permet de remédier à ce problème. Cette approche consiste à établir des alignements non pas entre les mots, mais entre les segments ou séquences de mots. L'unité de traduction devient alors la séquence de mots. Ceci est illustré par la figure 1.2 où les phrases allemandes et anglaises sont d'abord segmentées en séquences de mots et ensuite alignées ensemble.

Dans ce qui précède, il est à noter que l'approche **PBT** suppose implicitement que les unités de traduction sont des mots ou des séquences de mots qui sont encore plus grandes que le mot. Ceci pourrait poser un autre problème surtout pour les langues où les plus petites unités portant un sens sont lexicalement plus petites que le mot. Ces unités, qui

sont plus petites que le mot et qui chacune d'elles porte un sens à elle seule, sont appelées morphèmes (Matthews, 1991). La complexité morphologique d'une langue est définie comme le degré d'utilisation de ces morphèmes pour exprimer, entre autres, des relations grammaticales. Avec un faible ratio de morphèmes par mot, l'anglais est typologiquement classé comme une langue morphologiquement pauvre (ou limitée). Pour une telle langue, l'expression des relations syntaxiques repose largement sur l'ordre des mots (Clifton, 2010).

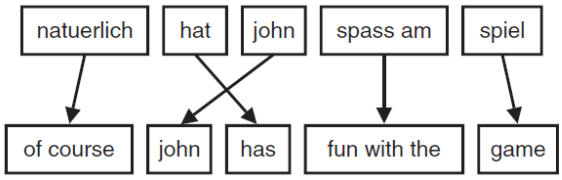

Figure 1.2 : Alignement entre séquences ou segments de mots allemands et anglais (Koehn et Ltd, 2010).

Les langues qui possèdent un ratio élevé de morphèmes par mot sont considérées comme des langues morphologiquement riches. En effet un mot unique dans ces langues peut contenir un nombre élevé de morphèmes ce qui exprime une richesse informationnelle. Le finnois ou l'inuktitut en sont des exemples. Pour mieux comprendre, prenons le mot inuktitut ‘‘*qaisaaliniaqquunngikkaluaqpuq*’’. Ce dernier est formé par les 7 morphèmes qai|saali|niaq|quu|nngik|kaluaq|puq. De telles langues (où le mot est composé par plusieurs morphèmes) sont dites agglutinatives. Elles sont caractérisées par un vocabulaire de taille importante. En effet les différentes combinaisons de morphèmes peuvent engendrer différents mots ayant un sens particulier, formant ainsi un vocabulaire plus riche. Par contre, les langues, dont le ratio de morphèmes par mot est faible, sont répertoriées comme des langues morphologiquement pauvres ou limitées. Le vocabulaire de telles langues est relativement petit par rapport aux langues morphologiquement riches.

Ainsi en présence d'une langue morphologiquement riche, que ce soit du côté source ou du côté cible, les approches **PBT** et **WBT** présentent des problèmes du fait qu'elles ne considèrent pas les morphèmes comme des unités atomiques de traduction.

1.1 Motivation

La complexité morphologique constitue un problème réel pour les systèmes de traduction automatique **SMT**. Cette complexité présente un défi à relever par la linguistique computationnelle. Ceci révèle les limites des modèles de traduction automatique **SMT** en l'absence d'une formulation appropriée des aspects morphologiques. L'introduction de la dimension complexité morphologique dans ces modèles permet de réduire l'erreur de mauvaise spécification et de renforcer, par conséquent, leur efficacité et leur emploi, particulièrement, dans la traduction des langues à morphologie riche. Le développement de méthodes et d'outils susceptibles de remédier aux limites des modèles de traduction traditionnels constitue la motivation essentielle de ce mémoire.

1.1.1 Sous-performance des approches traditionnelles et nécessité d'extension

Un nombre grandissant de travaux porte sur l'application de l'approche **PBT** pour des tâches de traduction comportant au moins une langue morphologiquement riche comme source ou cible. Les résultats de ces travaux ne sont pas concluants en termes de qualité de traduction. Dans son travail de pionnier, (Koehn, 2005) a pu tester la performance de l'approche **PBT** à travers la conception de 110 systèmes de traductions automatiques correspondant à 110 paires de langues. Le développement de ces systèmes est fait à partir des traductions de 11 langues européennes. Pour chaque paire, l'entraînement est effectué à l'aide du corpus Europarl v3[1] qui est une collection de textes parlementaires relatifs au parlement européen. Le corpus contient près d'un million de phrases pour chacune des 11 langues européennes retenues. La performance d'un système de traduction donné est mesurée à l'aide de la métrique **BLEU** (Bilingual Evaluation Understudy). **BLEU** est une mesure introduite par (Papineni et al., 2002) et qui sert à évaluer la qualité de la traduction obtenue. Elle est considérée comme la mesure la plus utilisée et la plus appropriée pour

[1] http://www.statmt.org/europarl/v3/

cette tâche. La popularité de cet indicateur provient de sa forte corrélation avec l'évaluation humaine (Papineni et al., 2002). Un score **BLEU** peut prendre une valeur entre 0 et 100 où 100 correspond à une traduction identique à la référence. Des valeurs élevées du score **BLEU** traduisent des niveaux de qualité de traduction meilleurs. Une discussion plus détaillée de la métrique ainsi que de ses propriétés sera donnée ultérieurement dans la section 2.3.1.

Les résultats de l'étude expérimentale, menée par (Koehn, 2005), montrent que la meilleure performance de traduction des systèmes **PBT** est obtenue pour le système relatif à la paire de langues espagnol-français (l'espagnol étant la source et le français étant la cible) avec un score de 40.2. Pour d'autres paires de langues, les systèmes de traduction **PBT** ont été moins performants, mais les résultats enregistrés restent, quand même, significatifs. Pour les paires français-anglais et allemand-anglais les scores **BLEU** sont respectivement 30.0 et 25.3, alors que le score le plus bas soit 10.3 est attribué à la paire néerlandais-finnois. D'ailleurs, le finnois, comme il a été déjà précisé, est une langue morphologiquement riche et son vocabulaire est le plus grand parmi les 11 langues.

Par ailleurs, un autre résultat qui ressort de l'étude de (Koehn, 2005) est l'existence d'une corrélation négative entre la complexité ou la richesse morphologique et la qualité de traduction reportée par le score **BLEU**. En d'autres termes lorsque la taille du vocabulaire (nombre de mots distincts) d'une langue devient de plus en plus importante, le score de traduction devient de plus en plus faible indiquant une tendance vers la détérioration de la qualité de traduction et donc de la performance de l'approche adoptée. Ce comportement est illustré dans la figure 1.3. L'axe des abscisses indique la taille des vocabulaires associés à 10 langues européennes et l'axe des ordonnées porte le score **BLEU** obtenu en traduisant à partir de ces langues vers l'anglais. Ici l'anglais n'est pas indiqué. Le score le plus bas correspond à la langue qui contient le vocabulaire le plus important (de plus grande taille), ici c'est le finnois.

La qualité de traduction d'une langue vers une autre semble, aussi, être tributaire de la position qu'occupe la langue morphologiquement riche dans le couple objet de traduction.

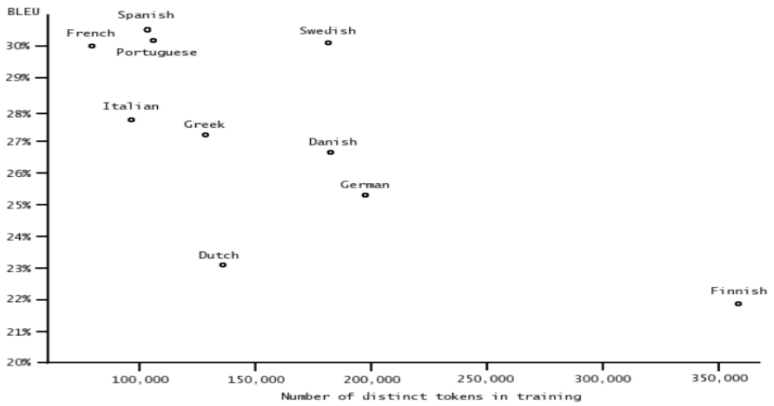

Figure 1.3 : Qualité de traduction et richesse morphologique (Koehn, 2005)

Le résultat est différent selon que la langue morphologiquement riche est d'un côté ou de l'autre de la cible. (Koehn, 2005) montre que si la langue morphologiquement riche est du côté de la cible, la traduction devient de plus en plus difficile comme le montre le tableau 1.1. En effet on peut facilement voir que le score **BLEU** relatif à la traduction du finnois vers l'anglais est nettement supérieur au score **BLEU** relatif à la traduction de l'anglais vers le finnois. C'est le problème d'asymétrie de la source et de la cible.

Système de traduction	**BLEU**
Finnois-anglais	21.8
Anglais-finnois	13.0

Tableau 1.1 : Qualité des traductions relatives à la paire (finnois, anglais).

(Luong et al., 2010) expliquent que ceci est dû au fait que la langue source qui est morphologiquement pauvre ne contient pas les caractéristiques morphologiques de la langue cible ce qui rend la génération de celle-ci plus difficile.

Les limites de l'approche « Phrase-Based Translation ou **PBT** » et leurs effets sur la détérioration du niveau de la qualité de traduction montrent son inadéquation pour la tâche

de traduction incluant une langue morphologiquement riche et surtout lorsque celle-ci est du côté de la cible.

1.2 Approches adoptées pour la traduction automatique comportant une langue à morphologie riche

Les limites identifiées par l'étude du problème de traduction statistique vers une langue morphologiquement riche sont des défis réels à relever et constituent l'une des principales motivations de notre recherche. Dans ce contexte plusieurs méthodes ont été proposées. Il ne s'agit pas ici d'en faire une étude exhaustive, mais, par contre, nous retenons deux méthodes récentes qui serviront de support pour les développements ultérieurs, proposés dans ce mémoire. La section suivante est consacrée à la discussion des types de modèles relatifs à ces deux méthodes. Le premier type comporte des modèles de traduction à base de facteurs « Factored Translation Models » et le second appréhende les modèles de traduction à base de segmentation « Segmented Translation Models ».

1.2.1 Les modèles de traduction à base de facteurs

Pour pallier aux insuffisances relevées, par l'application des méthodes **PBT,** d'autres approches ont été proposées. Dans ce contexte, les premières tentatives remontent aux travaux de (Yang et Kirchhoff, 2006). Ces derniers proposent un modèle de traduction qui permet de représenter l'information relative aux différents niveaux de l'analyse morphologique. Cette représentation a l'avantage de permettre, entre autres, le calcul d'un certain nombre de statistiques correspondant à des formes de surface observées à l'entraînement . Les formes de surface sont les formes de mots telles qu'observées dans un texte. (Nous expliquerons en quoi consiste la phase d'entraînement ultérieurement dans la section 2.1). Le but ultime d'une telle approche est de pouvoir traduire, à partir des formes les plus générales ayant servi à l'apprentissage du modèle de traduction, les formes de mots inconnus et non encore observés.

Dans la même lignée (Koehn et Hoang, 2007) proposent une nouvelle approche appelée « Factored Translation (**FT**) » qui est basée sur le même principe que celui de (Yang et Kirchhoff, 2006). Cette approche consiste à exploiter l'information morphologique au niveau du mot. De manière plus précise, au lieu de traduire les formes de mots, la tâche consiste à traduire l'information morphologique associée au mot et à générer ensuite le mot dans la langue cible à partir de l'information traduite.

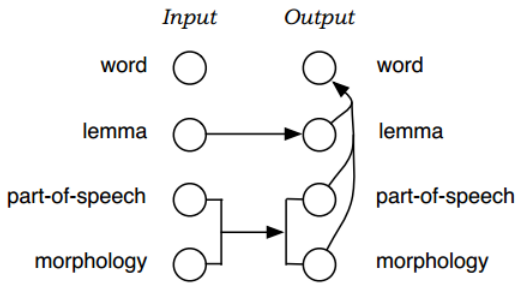

Figure 1.4 : Exemple d'une modélisation possible de l'information morphologique en utilisant l'approche **FT** (Koehn et Hoang, 2007).

L'information morphologique associée au mot est généralement modélisée sous la forme d'un vecteur de caractéristiques. Dans l'exemple de la figure 1.4, le vecteur de caractéristiques est représenté par 3 composants (le lemme, la catégorie grammaticale, la morphologie). Dans cet exemple, la traduction est établie en 3 étapes :

1. La traduction de lemmes.
2. La traduction de la paire catégorie grammaticale et morphologie.
3. La génération de la forme du mot à partir des traductions.

Ainsi, des formes de mots inconnus peuvent être générées grâce à l'information morphologique ajoutée. À titre d'illustration, supposons qu'on ait à traduire un texte vers l'anglais. De plus supposons que le mot anglais ‘‘*houses*’’ n'a pas été observé au cours de l'entraînement, et que les formes morphologiques plurielles (*plural*) associées aux noms communs (*NN*) en anglais génèrent un s à la fin du mot. Dans ces conditions, l'application

du principe des trois étapes précédentes peut générer la forme du mot ‘‘*houses*’’ dans le vecteur des caractéristiques (‘‘*house*’’,*NN*,*plural*).

Les modèles **FT** ont fait aussi l'objet des travaux plus récents, portant sur la traduction d'une langue morphologiquement pauvre vers une langue morphologiquement riche, comme ceux de (Clifton, 2010). La particularité de ces derniers travaux, qui consistaient en une traduction de l'anglais vers le finnois, est la nature même des vecteurs de caractéristiques correspondant à la fois à la langue cible et à la langue source. En effet du côté de la cible (le finnois) les caractéristiques prises sont le mot lui-même, le stemme (partie du mot ne contenant pas le suffixe) et le suffixe. Tandis que du côté de la source (l'anglais), les caractéristiques utilisées sont le mot lui-même et la catégorie grammaticale (Part Of Speech). Les résultats obtenus par (Clifton, 2010) montre la supériorité de l'approche **PBT** prise comme benchmark (référence ou base de comparaison) par rapport à l'approche **FT**. (Clifton, 2010) indique que la sous-performance des modèles **FT**, dans ce cas, est vraisemblablement due à la complexité morphologique de la langue finnoise qui rend difficile la tâche d'apprentissage et de mémorisation de l'information morphologique représentée par les combinaisons potentielles des stemmes et suffixes finnois. Un autre facteur expliqué par (Clifton, 2010) est l'inefficacité des mécanismes de génération incorporés dans les modèles **FT**. En effet, la génération se fait au niveau des mots et non pas au niveau des séquences de mots. Ainsi, la sortie ne tient pas compte des dépendances entre les morphèmes distants et on aboutit donc à une sortie non fluide.

1.2.2 Les modèles de traduction à base de segmentation

L'amélioration de la qualité de traduction a continué de susciter un intérêt particulier chez les chercheurs dans le domaine de la linguistique computationnelle. À cet effet, le nombre de travaux qui prennent en considération l'information morphologique sous différents aspects et de manières différentes n'a cessé d'augmenter. Les recherches menées par (Clifton et Sarkar, 2011), (Luong et al., 2010), (Nguyen et al., 2010) et (Chung et Gildea, 2009) s'insèrent dans le cadre de cette orientation méthodologique.

(Clifton et Sarkar, 2011) utilisent les modèles de traduction à base de segmentation ou « Segmented Translation Models (**STM**) » pour accomplir des tâches de traduction sur des données ayant des structures morphologiques complexes. Ils proposent dans leur expérimentation l'utilisation préalable de méthodes de segmentation supervisée et non supervisée pour entrainer des modèles à base de séquences d'unités lexicales (les modèles **PBT**). Une méthode de segmentation permet de segmenter les mots en des morphèmes. Cette combinaison de modèles de segmentation et de modèles de traduction automatique remonte aux travaux de (Nguyen et al., 2010) dont l'objet est d'augmenter le degré de symétrie entre la cible et la source. Les travaux de (Chung et Gildea, 2009) vont dans le même sens, mais avec l'application de plusieurs types de segmentations.

La méthode proposée par (Clifton et Sarkar, 2011) consiste à établir un prétraitement qui sert à décomposer les mots du vocabulaire finnois en des segments à l'aide de l'outil de segmentation non supervisée **Morfessor** (Creutz et Lagus, 2005). **Morfessor**[2] construit un vocabulaire de segments afin de représenter n'importe quel mot à partir d'une concaténation de segments. Ceci donne la possibilité d'opérationnaliser la segmentation, d'obtenir le vocabulaire de segments optimaux et de résoudre, en partie, le problème de la rareté des données. Les résultats déclarés dans l'étude expérimentale de (Clifton et Sarkar, 2011) montrent une supériorité de la qualité de traduction par rapport à la méthode de traduction **Phrase-Based** (**PBT**) prise comme référence. Le score **BLEU** du système de référence étant de 14.68, ces derniers ont pu améliorer la qualité de la traduction en obtenant un score **BLEU** de 15.09.

(Clifton et Sarkar, 2011) ont réussi aussi à battre le système de traduction **Phrase-Based** (**PBT**) de référence une deuxième fois en adoptant une approche similaire à la précédente. Le système de traduction qu'ils préconisent permet dans un premier temps de traduire de l'anglais vers des formes de surface du finnois ne contenant pas de suffixes. La deuxième étape consiste alors à prédire ces suffixes. **Morfessor** a été employé afin de déduire les suffixes qui devaient être supprimés des mots utilisés pour entraîner le système

[2] http://www.cis.hut.fi/projects/morpho/morfessorcatmapdownloadform.shtml

de traduction. Les champs aléatoires conditionnels ou « Conditional Random Fields (**CRF**) » sont utilisés pour la prédiction des suffixes des mots. Avec une telle approche, (Clifton et Sarkar, 2011) ont obtenu un score **BLEU** de 14.87.

Pour la même tâche de traduction de l'anglais vers le finnois, (Luong et al., 2010) ont réussi à battre le système **PBT** de référence par 0.58 de **BLEU** d'écart (référence 14.08 (Luong et al., 2010) 14.58 **BLEU**) en suggérant une représentation hybride mot-morphème, où l'unité atomique de traduction est le morphème. Le respect de la forme des mots a été obligatoirement maintenu à tous les stades du processus de traduction. Le respect de cette contrainte à tous les niveaux a permis au système de traduction préconisé de générer comme forme de surface, des mots et non pas des morphèmes incomplets. Les mots du vocabulaire ont été préalablement segmentés à l'aide de **Morfessor**.

L'une des limites, qu'on peut relever au niveau des systèmes élaborés par (Clifton et Sarkar, 2011) et (Luong et al., 2010), se rapporte à l'outil de segmentation non supervisée **Morfessor** lui-même. En effet, ces derniers ne prennent en compte que l'information monolingue relative à la langue contenant le vocabulaire à segmenter. D'autres approches ont traité le problème de la traduction incluant au moins une langue morphologiquement riche tout en élaborant une méthode de segmentation qui tient compte de l'information bilingue.

Dans cette optique, (Nguyen et al., 2010) ont développé une méthode d'apprentissage non supervisée permettant de segmenter le texte de la langue morphologiquement riche. La segmentation des morphèmes, ainsi proposée, est établie d'une manière qui permet de construire l'alignement le plus probable entre les unités des deux langues. Pour capturer le contenu informationnel bilingue (Nguyen et al., 2010) proposent un modèle bayésien génératif. La génération est réalisée en deux étapes. Un modèle de segmentation monolingue génère tout d'abord la segmentation de la phrase source. En seconde étape, un modèle d'alignement effectue l'alignement entre les segments de la phrase source et les mots de la phrase cible. Les paramètres régissant ce processus de génératif sont appris sur les données d'entraînement.

L'approche précédente de (Nguyen et al., 2010) pose un problème particulier. En effet pour pouvoir segmenter du texte et utiliser l'identité précédente, nous avons besoin impérativement de la traduction qui lui correspond afin de pouvoir déterminer la segmentation optimale. Cette traduction n'existe évidemment pas pour de nouvelles données qu'on voudrait traduire. Nous sommes dans la situation de données manquantes à restituer.

Pour pouvoir déterminer la distribution des alignements manquants, les auteurs proposent une méthode d'inférence qui utilise un outil de désambigüisation morphologique propre à la langue à segmenter et qui permet de générer un échantillon de morphèmes. Pour certaines langues, de telles ressources sont rares. Néanmoins, (Nguyen et al., 2010) ont réussi à obtenir de légères améliorations dans des tâches de traduction de l'arabe vers l'anglais (référence 54.00 , (Nguyen et al., 2010) 56.82).

De plus, comme (Clifton et Sarkar, 2011) l'ont indiqué, l'idéal serait de poser le problème de traduction incluant une langue à morphologie riche d'une manière générique. Cela permet de concevoir des solutions qui peuvent être appliquées sur n'importe quelle tâche de traduction incluant au moins une de ces langues.

Traitant toujours de la capture de l'information bilingue, (Chung et Gildea, 2009) ont développé une méthode de segmentation non supervisée dont le principe consiste à apprendre la segmentation des mots du vocabulaire d'une langue à morphologie riche à partir de l'alignement établi par le modèle IBM 1 entre les deux langues. (Nous verrons plus tard, dans la section 2.1, les modèles IBM). La segmentation la plus probable est produite à l'aide de l'algorithme de Viterbi. Des améliorations de la qualité de la traduction ont été recensées suite à l'application de cette méthode pour la traduction du coréen et du chinois vers l'anglais.

1.3 Notre approche : Objet et contributions attendues

1.3.1. Objet du mémoire

Le travail dans ce mémoire s'insère dans le cadre des efforts qui consistent à introduire l'information morphologique dans la spécification des modèles de traduction automatique **SMT**. Notre approche est similaire en philosophie, à celles de (Chung et Gildea, 2009), et (Clifton et Sarkar, 2011), mais diffère, de façon significative, quant aux algorithmes de segmentations proposés et à la nature même du schéma de combinaison de ces algorithmes avec des modèles statistiques de traduction automatique à base de phrases **PBT**. De manière plus concrète, notre approche consiste à tirer profit de l'information bilingue pour poser le problème de traduction de manière générique, c'est-à-dire, non spécifique à une langue particulière donnée, et établir la segmentation des mots pour des langues morphologiquement riches. Pour accomplir cette tâche, nous proposons un nouvel algorithme où la segmentation des mots du vocabulaire de la langue cible est établie à partir des valeurs des probabilités de traduction de mots estimées par le logiciel **Moses**[3] (Koehn et al., 2007) (nous verrons plus tard, dans la section 2.1, comment les probabilités de traduction de mots sont estimées). Cet algorithme de segmentation est facilement généralisable à d'autres langues. Nous nous basons sur l'approche à base de séquences **PBT** implémentée dans **Moses** pour la conception de tous les systèmes de traductions testés dans ce mémoire.

1.3.2. Résultats escomptés et esquisse du contenu du mémoire.

Les résultats attendus dans ce travail peuvent être énumérés dans la liste qui suit :

1. Apporter des solutions alternatives simples aux problèmes posés par la traduction automatique des langues à morphologie complexe qui soient généralisables.

[3] http://www.statmt.org/moses/

2. Contribuer à l'élimination partielle du problème de rareté des données en proposant une segmentation du vocabulaire de la langue à morphologie complexe en faisant usage de l'information bilingue à tous les stades du processus de traduction.
3. Contribuer à l'amélioration de la qualité de traduction en intégrant dans le système de traduction l'information procurée par la segmentation.

Le reste de ce mémoire est organisé de la façon suivante. Le chapitre 2 adresse le problème de traduction dans le cadre de deux grandes références de modèles statistiques. Ce sont les modèles à base de mots et baptisés modèles **WBT ou «** Word Based Translation Models **»** et les modèles à base de séquences désignés par **PBT**. Ces modèles constituent en fait les composantes de base dans l'architecture des systèmes de traductions automatiques que nous développons et implémentons tout au long de ce mémoire de recherche. Les étapes d'apprentissage (entraînement), de développement et de décodage et d'évaluation de ces modèles sont introduites.

Le chapitre 3 prend une orientation plus pratique. Il porte sur l'implémentation d'un système statistique de traduction automatique de base. Plus particulièrement il s'agit de la traduction de l'anglais, considéré comme une langue à morphologie pauvre, vers l'inuktitut qui est une langue à morphologie complexe. Le choix de l'inuktitut est légitimé sur le plan théorique par le fait que c'est une langue qui fait réunir toutes les difficultés et les lacunes que peut rencontrer un système de traduction automatique **SMT**. Le système de traduction de référence que nous proposons pour effectuer la traduction automatique de l'anglais vers l'Inuktitut est basé sur l'approche **PBT**. Les résultats de l'expérimentation montrent une performance appréciable du système de traduction retenu. Un examen plus détaillé des données a révélé que la bonne qualité des résultats de traduction obtenus est imputable à la facilité d'apprentissage (surapprentissage ou « overfitting ») du corpus inuktitut-anglais. Ceci provient vraisemblablement du fait que les données d'évaluation sont proches des données observées lors des phases d'entraînement et de développement.

À cause du manque de données dans la langue inuktitute, nous avons alors décidé d'étudier, dans le chapitre 4, le problème de traduction vers une langue à morphologie riche

en considérant essentiellement le finnois comme langue cible et pour lequel les données d'entraînement et de développement dont on dispose sont différentes des données d'évaluation. Cette langue fait, comme nous l'avons mentionné plus haut, l'objet de nombreuses études, ce qui nous permet de situer l'apport de notre approche à l'état de l'art en **SMT.**

Le chapitre 4 traite, dans une première étape, des fondements théoriques des systèmes statistiques de traduction automatique susceptibles de saisir les diversités des structures morphologiques et éviter par conséquent les obstacles posés par la complexité morphologique et dénombrés dans les chapitres précédents. L'architecture de ces systèmes comporte, entre autres, deux composantes de traitement des données, l'un en amont que nous appelons prétraitement, l'autre en aval que nous appelons post-traitement. Le processus de traduction proposé est appliqué alors pour la traduction de l'anglais vers le finnois, mais aussi de l'anglais vers l'inuktitut. L'utilité des opérations de prétraitement suggérées (stemming, segmentation, etc.) réside dans le fait qu'ils concourent pour la réalisation d'un apprentissage de la structure morphologique des mots. En effet, partant de cet apprentissage il devient facile de construire un vocabulaire réduit à partir duquel on peut générer n'importe quel mot en concaténant les segments faisant partie du vocabulaire. Les transformations préalables requièrent des opérations de conversions à postériori (accolage des segments, ou désambigüisation, etc.) qui leur sont, dans la plupart des cas, fortement reliées. Ces reconversions permettent de rétablir le finnois généré par les systèmes de traduction en finnois correct.

Pour la tâche de traduction de l'anglais vers l'inuktitut, nous appliquons un seul prétraitement (le stemming) pour vérifier si les résultats obtenus pour la tâche de traduction anglais-finnois concordent avec celle de l'inuktitut.

Le second volet couvert dans ce chapitre revêt une forme pratique consistant à la validation expérimentale des systèmes de traduction proposés. La lecture des résultats révèle la supériorité du modèle de langues 5-grammes (voir section 2.1). De là, il n'est pas difficile de conclure que la réduction du vocabulaire, par les opérations de stemming, ne

pourrait pas assurer, à elle seule, la conservation de l'information morphologique des mots finnois et inuktituts. Dans ce cas d'espèce, le choix des formes fléchies devient de plus en plus compliqué, rendant ainsi l'opération de désambigüisation plus difficile à accomplir. En fait, les meilleurs résultats (affichés par les scores **BLEU, SER,** et **WER**) sont notés pour les processus de traduction qui utilisent la segmentation comme outil préalable de prétraitement. En effet, la meilleure génération morphologique à l'aide de l'algorithme **SRILM Disambig** (vois section 4.2.4.2) est assurée lorsque les données sont segmentées. La segmentation pourrait être perçue dans ce cas, à la fois, comme un moyen de réduction de la taille du vocabulaire et de conservation de l'information morphologique.

Malgré les bons résultats notés en présence de données segmentées, la primauté de la segmentation en tant qu'outil de prétraitement ne peut être garantie à priori. Les comparaisons entre systèmes de traduction, en présence de données segmentées, ne peuvent se faire, de manière intrinsèque, qu'après avoir restitué les mots du corpus dont les unités ont été préalablement segmentées. Cette opération est assurée par des transformations d'accolage des segments. L'application de ces transformations aux données relatives au corpus, antérieurement segmenté par (Clifton et Sarkar, 2011), ont permis de restituer les mots de ce corpus. Ces données ont servi à l'apprentissage du processus de traduction référence. Les scores **BLEU**=14.97**,** et **WER**=63.36 enregistrés par ce système référence montrent sa supériorité. Ce résultat concorde avec celui établi par (Luong et al., 2010) et qui stipule que fait de considérer les morphèmes comme unités atomiques de traduction permet d'améliorer la qualité de l'alignement des mots, mais ceci est insuffisant pour améliorer la traduction. Des méthodes permettant la conservation de la forme des mots doivent, donc, être développées et appliquées à tous les stades du processus de traduction. Nos résultats ne concordent pas avec ceux de (Clifton et Sarkar, 2011) qui affirment qu'une segmentation utilisant l'information morphologique monolingue de la langue permet à elle seule d'améliorer la traduction.

Des améliorations de la qualité de la traduction dans des tâches incluant une langue à morphologie riche font l'objet des développements que nous proposons au chapitre 5.

Dans ce cadre, un nouvel algorithme de segmentation est conçu. La segmentation des mots du vocabulaire de la langue cible est réalisée à partir de la distribution des probabilités de traduction lexicales estimées par **Moses**. Nous faisons usage de cette distribution comme information bilingue pour la segmentation du vocabulaire finnois. Pour la conception de tous les systèmes de traduction testés ici nous nous appuyons sur l'approche à base de séquences d'unités lexicales **PBT** implémentée dans **Moses.**

Ici, nous avons tenu à ce que les aspects théoriques relatifs au paradigme de segmentation et le pseudo-code de l'algorithme sous-jacent soient décrits avec précision. La performance de cet algorithme est évaluée à partir d'un échantillon de 10000 mots finnois distincts et de leurs traductions obtenues à l'aide de **Google Translator Toolkit**. L'évaluation de la qualité de la traduction produite par notre algorithme de segmentation par rapport à celle de **Google Translator Toolkit** est due au fait que l'outil mentionné est considéré comme une référence pour la traduction automatique. Comme on ne possède pas de référence attestée et que **Google Translator Toolkit** produit des traductions de mots qui sont souvent bonnes, il nous a semblé judicieux de recourir à l'utilisation des références Google et ce en dépit des problèmes liés au côté perfectible de celle-ci. L'échantillon des 10000 observations est réalisé, par tirage aléatoire, à partir du vocabulaire d'entraînement du corpus finnois**.**

Notre algorithme a été appliqué aussi pour effectuer la segmentation du corpus finnois. L'évaluation de la performance de notre algorithme par rapport à d'autres algorithmes de segmentation a fait l'objet d'une analyse comparative. Dans cet exercice le même corpus finnois est présenté à tous les algorithmes qui concourent. Les résultats de cette compétition assurent la supériorité de notre approche, malgré le fait que l'approche de (Clifton et Sarkar, 2011) permet de réduire davantage le vocabulaire finnois. En effet l'écart observé, en termes des scores **BLEU**, entre la performance (Clifton et Sarkar, 2011) et la nôtre est de **0,28**. Cette supériorité peut s'expliquer par la particularité des procédures mises en place dans notre schéma de segmentation et qui permettent une meilleure restitution des structures morphologiques bilingues. En effet l'inclusion de l'information bilingue permet au système de traduction d'aligner les segments finnois avec les mots

anglais d'une manière plus précise. Par ailleurs, en comparant avec l'approche de traduction de référence (ou **baseline**), les résultats reflètent une meilleure qualité de traduction pour notre approche. L'écart observé entre le score **BLEU** des deux approches est négligeable dans ce cas.

Le dernier chapitre est consacré à la conclusion qui résume le travail établi dans ce mémoire et les principales contributions.

Chapitre 2 Les modèles statistiques de traduction automatique : spécification, apprentissage, développement et évaluation

Dans ce chapitre on s'intéresse aux modèles statistiques de traduction automatique à base de mots **WBT** ainsi qu'aux modèles à base de séquences de mots **PBT.** Ces modèles constituent en fait les composantes de base dans l'architecture des systèmes de traductions automatiques que nous développons et implémentons tout au long de ce mémoire de recherche. Plusieurs spécifications sont approchées. Les phases d'apprentissage (entraînement), de développement et de décodage de ces modèles font l'objet des discussions des sections 2.1 et 2.2. La section 2.3 est dédiée à la présentation des métriques ou critères d'évaluation de la performance de ces modèles et de la qualité des traductions qu'ils produisent.

2.1 Principe général des modèles statistiques de traduction

Le principe des systèmes de traduction est de produire la meilleure traduction $\mathbf{e}_{\text{best}}$ parmi toutes les phrases **e** de l'ensemble **E** potentiellement infini de phrases qui peuvent être créées à partir de l'alphabet de la langue cible**.** La phrase **f**, à traduire, fait partie de l'ensemble **F** potentiellement infini de phrases qui peuvent être créées à partir de l'alphabet de la langue source**.** En employant la règle de Bayes, on obtient :

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}} \frac{p(\mathbf{f}|\mathbf{e}).p(\mathbf{e})}{p(\mathbf{f})} = \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e}).p(\mathbf{e}) \qquad (2.1)$$

Ici, $\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f})$ exprime l'ensemble des phrases **e** permettant d'obtenir une valeur maximale de la probabilité conditionnelle $p(\mathbf{e}|\mathbf{f})$ (probabilité de **e** sachant **f**). L'égalité de l'équation à (2.1), où le dénominateur est ignoré, tient au fait que $p(\mathbf{f})$ est constante pour le besoin de maximisation.

Un modèle de langue $p(\mathbf{e})$ est intégré dans le modèle de traduction. Un modèle de langue assure la cohérence ou la fluidité de la sortie en calculant la probabilité d'occurrence

d'une phrase **e** dans un texte monolingue écrit dans la langue de **e**. A cette fin, la notion de n-grammes est alors utilisée. Un n-grammes est une séquence de n mots consécutifs $e_k . e_2 \dots e_{n+k-1}$ où e_i désigne le $i^{\text{ème}}$ mot de la phrase e de taille m. La probabilité d'occurrence $p(\mathbf{e})$ peut se calculer à partir des probabilités des n-grammes qui se trouvent dans un texte monolingue.

$$p(e_1 . e_2 \dots e_m) = \quad p(e_1)p(e_2|e_1) \prod_{\mathbf{i=3}}^{\mathbf{m}} p(e_i|e_1 \dots e_{i-1}) \cong$$

$$p(e_1)\, p(e_2|e_1) \prod_{\mathbf{i=3}}^{\mathbf{m}} p(e_i|e_{i-n+1} \dots e_{i-1}) \qquad (2.2)$$

$p(e_i|e_1 \dots e_{i-1})$ exprime la probabilité conditionnelle que la séquence de mots consécutifs $e_1 \dots e_{i-1}$ soit suivie par le mot e_i. $\prod_{\mathbf{i=3}}^{\mathbf{m}} p(e_i|e_{i-n+1} \dots e_{i-1})$ est une approximation markovienne d'ordre n-1 du produit $\prod_{\mathbf{i=3}}^{\mathbf{m}} p(e_i|e_{i-n+1} \dots e_{i-1})$. Un modèle n-gramme fait donc l'hypothèse que la prédiction d'un mot d'indice i dans une séquence dépend seulement des n-1 mots qui le précède, c'est-à-dire, une hypothèse markovienne d'ordre n-1. Une manière d'estimer ces probabilités est de calculer les fréquences relatives aux n-grammes qui se trouvent dans un corpus monolingue. Illustrons l'exemple d'un trigramme (ou 3-gramme) en admettant que **e** contient n mots. Nous aurons alors :

$$\mathbf{p(e)} = p(e_1 e_2 \dots e_m) = p(e_1).p(e_2|e_1) \dots p(e_m|e_{m-2}e_{m-1}) \qquad (2.3)$$

$$\text{et} \qquad p(e_i|e_{i-2}e_{i-1}) = \frac{compte(e_{i-2}e_{i-1}e_i)}{\sum_{\mathbf{e'_i}} compte(e_{i-2}.e_{i-1}.e'_i)} \qquad (2.4)$$

Où $compte$ désigne le nombre d'occurrences d'un n-grammes dans le corpus monolingue. $\sum_{\mathbf{e'_i}} compte(e_{i-2}.e_{i-1}.e'_i)$ désigne la somme des comptes sur tous les trigrammes contenant $e_{i-2}\, et\, e_{i-1}$ comme étant les deux premiers mots de la séquence. D'autres techniques de lissages tels que (Chen et Goodman, 1996) et (Kneser et Ney, 1995) sont employées afin de contourner le problème des séquences de n-grammes de compte nul dans le corpus monolingue. En effet, l'estimée de la probabilité d'une séquence n-gramme sera nulle dès lors que la séquence n'a pas été rencontrée dans le corpus d'entraînement.

2.2 Modèles de traduction à base de mots et alignements de mots

En général, la tâche de traduction entre deux langues présuppose l'existence d'un ensemble informationnel formé de deux textes dont chacun constitue la traduction de l'autre et dont les phrases sont alignées. Un tel ensemble de textes est appelé corpus parallèle. Le problème dans les corpus de textes parallèles est que l'alignement de mots n'est habituellement pas fourni malgré le fait que les phrases soient alignées. Par exemple, lorsque la phrase anglaise ''*He did it again.*'' est alignée avec la phrase française ''*Il l'a fait encore une fois.*'', l'alignement de mots est manquant alors que l'alignement des phrases est assuré dans ce cas précis. En fait, puisque les mots ne sont pas annotés, il est difficile de savoir quel mot est considéré comme une traduction de l'autre. Si les alignements de mots étaient disponibles, les probabilités lexicales pourraient être déduites en comptant le nombre de fois où un mot f est aligné avec e dans les deux textes. Nous obtenons ainsi la probabilité de traduction lexicale :

$$p(f|e) = \frac{compte_e(f)}{\sum_{f'} compte_e(f')} \tag{2.5}$$

Dans l'équation précédente; $compte_e$ désigne le nombre de fois où un mot f est aligné avec e dans les deux textes. $\sum_{f'} compte_e(f')$ désigne la somme sur tous les mots f' de la langue source, alignés avec le mot e.

L'équation (2.5) montre l'obligation de déterminer les alignements lorsque ces derniers ne sont pas pourvus, ce qui est typiquement le cas. L'alignement entre les mots est alors introduit à l'aide d'une variable cachée dans des modèles génératifs. On mentionne l'existence de nombreux travaux qui couvrent ce déficit dont les plus populaires sont connus sous le nom de modèles **IBM** (Brown, Pietra, Pietra, et Mercer, 1993)**.** Nous décrivons dans la suite ces modèles de traduction à base de mots en s'inspirant de la description donnée dans (Patry, 2010). L'idée maîtresse de ces modèles est d'introduire une variable cachée qui représente l'alignement entre la phrase source et la phrase cible. De tels modèles s'avèrent utiles puisqu'on peut déduire les probabilités lexicales (probabilité de traduction d'un mot étant donné le mot ou la phrase à traduire) et les probabilités

d'alignement qui en ressortent. Il existe 5 spécifications différentes de modèles **IBM** à base de mots et une spécification sous la forme d'un modèle de Markov caché ou « Hidden Markov Model (**HMM**) ». Nous nous concentrons, sur les deux premières puisqu'elles sont les plus connues dans la littérature. L'équation (2.6), donne la forme, ou la spécification, suggérée pour les modèles **IBM** en exprimant une histoire générative. Les probabilités reportées dans (2.6) constituent les paramètres inconnus (à estimer) de ces modèles.

$$p\left(\mathbf{f} \equiv f_1^{l_\mathbf{f}} \middle| \mathbf{e} \equiv e_1^{l_\mathbf{e}}\right) = \sum_{\mathrm{a} \in \mathrm{A}(\mathbf{e},\mathbf{f})} p\left(f_1^{l_\mathbf{f}}, \mathbf{a} \equiv a_1^{l_\mathbf{f}} \middle| e_1^{l_\mathbf{e}}\right) \quad (2.6)$$

Dans (2.6), la phrase $\mathbf{f} \equiv f_1^{l_\mathbf{f}}$ représente la phrase cible du modèle. $\mathbf{e} \equiv e_1^{l_\mathbf{e}}$ représente la phrase source.

$l_\mathbf{f}$ représente la longueur de la phrase **f** et $l_\mathbf{e}$ représente la longueur de la phrase **e**.

$f_1^{l_\mathbf{f}}$ est la séquence de mots contenant $l_\mathbf{f}$mots ($f_1\ , f_2\ , \dots , f_{l_\mathbf{f}}$).

$e_1^{l_\mathbf{e}}$ est la séquence de mots contenant $l_\mathbf{e}$ mots ($e_1\ , e_2\ , \dots , e_{l_\mathbf{e}}$).

$(\mathbf{e}, \mathbf{f}) \in (\mathbf{E}, \mathbf{F})$ **E** et **F** sont les ensembles infinis de phrases cibles et sources possibles à écrire. A$(\mathbf{e}, \mathbf{f})$ désigne l'ensemble des alignements entre **e** et **f.** Les alignements reconnus par ces modèles sont ceux qui associent à chaque mot de la traduction f_j, un et un seul mot.

$\sum_\mathrm{a}$ désigne la somme sur tous les alignements possibles **a**.

Un alignement **a** est équivalent à $a_1^{l_\mathbf{f}}$ qui est la séquence d'alignements ($a_1\ , a_2\ , \dots , a_{l_\mathbf{f}}$) entre les mots de **e** et les mots de **f** où a_j est un entier indiquant la position de la traduction relative au $j^{\text{ème}}$ mot de la phrase **f**. $a_j \in [0, l_\mathbf{e}]$, 0 étant la position associée de facto à un mot cible non associé à un mot source.

L'équation (2.7) donne la formule générale des modèles IBM 1 et 2 où :

$$p\left(f_1^{l_\mathbf{f}}, a_1^{l_\mathbf{f}} \middle| e_1^{l_\mathbf{e}}\right) = p(l_\mathbf{f} \mid \mathbf{e}) \prod_{j=1}^{l_f} p(a_j \mid a_1^{j-1}, f_1^{j-1}, l_\mathbf{f}, \mathbf{e}).p\left(f_j \mid a_1^j, f_1^{j-1}, l_\mathbf{f}, \mathbf{e}\right) \quad (2.7)$$

Où $p(l_\mathbf{f} \mid \mathbf{e})$ représente la probabilité conditionnelle de la longueur de **f.**

$p(a_j \mid a_1^{j-1}, f_1^{j-1}, l_\mathbf{f}, \mathbf{e})$ représente la distribution conditionnelle des positions des mots de **f** (probabilité conditionnelle de l'alignement relatif au $j^{\text{ème}}$ mot).

$p(f_j \mid a_1^j, f_1^{j-1}, l_{\mathbf{f}}, \mathbf{e})$ représente la distribution conditionnelle des mots de **f (**probabilité lexicale relative au $j^{\text{ème}}$ mot**).**

Pour le modèle **IBM 1**, les alignements sont supposés être équiprobables où $p(a_j \mid a_1^{j-1}, f_1^{j-1}, l_{\mathbf{f}}, \mathbf{e}) = \frac{1}{l_e+1}$. Le dénominateur est égal à $l_e + 1$ et non à l_e pour tenir compte pas des cas des mots cibles non associés à un mot source. Pour le modèle **IBM 2**, la distribution des alignements n'est plus uniforme, mais dépend de la position relative des mots alignés où $(a_j \mid a_1^{j-1}, f_1^{j-1}, l_f, \mathbf{e}) = p(a_j \mid j, l_{\mathbf{e}}, l_{\mathbf{f}})$. Le modèle **HMM** ressemble au modèle **IBM 2** sauf que la probabilité d'alignement est exprimée en fonction de la distance entre deux alignements successifs. Cette distance est obtenue par la soustraction des positions des traductions relatives à deux mots sources successifs appartenant à la phrase **f** $(a_j - a_{j-1})$:

$$p(a_j \mid a_1^{j-1}, f_1^{j-1}, l_f, \mathbf{e}) = p(a_j - a_{j-1}) \tag{2.8}$$

Les valeurs estimées des paramètres du modèle **IBM 1** servent comme valeurs initiales pour l'estimation des modèles **IBM 2** et **HMM**.

Il existe plusieurs types d'algorithmes statistiques pour l'estimation des paramètres d'un modèle. Mais comme les spécifications retenues ici expriment la probabilité de traduction en fonction des probabilités d'alignements qui sont dans la plus part des cas manquants et donc non observables, l'algorithme le plus recommandé pour l'estimation des modèles **IBM** est l'algorithme **EM** «Expected Maximisation » ou Espérance-Maximisation (Dempster, Laird, et Rubin, 1977). L'algorithme de Viterbi permet aussi de faire la même tâche (Viterbi, 1967). **EM** est un algorithme d'apprentissage de nature itérative qui permet de maximiser la vraisemblance des paramètres à apprendre. Le principe de son fonctionnement est décrit ci-dessous :

1. Initialiser le modèle avec des probabilités uniformes.
2. Ajuster le modèle sur les données disponibles.
3. Réajuster ou entrainer le modèle en calculant les nouvelles valeurs des paramètres.

4. Refaire les étapes 2 et 3 jusqu'à ce qu'il y ait convergence des paramètres.

Au cours des étapes précédentes, toutes les spécifications du modèle **IBM** sont initialisées par les valeurs obtenues à la convergence des modèles qui leur précèdent. Donc, le troisième modèle est initialisé par les valeurs obtenues par le deuxième modèle et le quatrième par les valeurs obtenues par le troisième et ainsi de suite.

Les modèles **IBM** 3, 4 et 5 sont plus complexes puisqu'ils introduisent de nouvelles spécifications qui traduisent d'autres aspects de traduction et rendent compte de plusieurs autres paramètres. Ces modèles sont plus couteux à entraîner et leur performance n'est que faiblement supérieure à celle du modèle HMM que nous venons de décrire. Le logiciel **Giza++ (Al-Onaizan et al., 1999)**, utilisé par le logiciel **Moses**, sur lequel, nous nous basons, dans ce mémoire, pour concevoir les systèmes de traductions, génère comme sortie, les valeurs apprises (estimées) par le modèle **IBM 4**.

Selon l'algorithme **EM** l'alignement le plus probable pour chaque mot de **e** est alors retenu. Chaque mot de **e** est ainsi associé au plus à un mot de **f.** Ceci permet d'obtenir des modèles non symétriques, ce qui pose un problème. Pour une meilleure qualité d'alignement, on devrait obtenir des modèles symétriques où un mot de **e** peut être aligné à plusieurs mots de **f**. Cette difficulté peut être contournée par le recours à une solution heuristique. L'heuristique la plus populaire consiste à entraîner les modèles IBM dans les 2 directions ($p(\mathbf{f}, \mathbf{a}|\mathbf{e})$ et $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$) et considérer l'alignement établi dans les deux directions. L'entraînement est effectué sur plusieurs étapes. La première consiste à générer l'intersection des alignements de mots. La deuxième étape consiste à ajouter les alignements de mots qui sont au voisinage des points d'intersection et qui font partie de l'union des deux alignements. Enfin, la troisième étape consiste à établir les alignements pour les mots qui n'ont pas été encore alignés. Ce processus d'alignement, connu sous le nom de symétrisation des alignements de mots, a été introduit par (Och et Ney, 2000). La figure 2.1 illustre le processus. **Moses** utilise le logiciel **Giza++** qui lui utilise le résultat établi par **IBM 4** pour effectuer le processus de symétrisation. Dans la figure 2.1, les carrés

noirs de l'alignement résultant représentent les points d'intersection entre les alignements et les points gris sont les points d'union qui ont été ajoutés.

Par ce procédé de symétrisation, nous avons l'assurance d'avoir des probabilités lexicales pour tous les mots dans le corpus et dans les deux sens de la traduction. Par exemple, dans la traduction de l'anglais vers le français ou l'inverse nous obtenons la distribution des probabilités lexicales dans les deux sens.

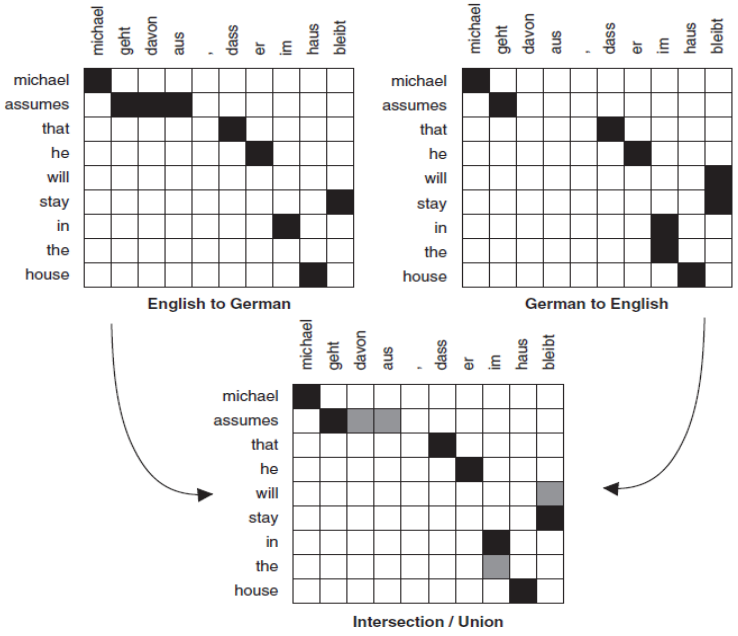

Figure 2.1 : Symétrisation des alignements de mots (Koehn et Ltd, 2010).

$$forme \begin{bmatrix} form & 0.63 \\ shape & 0.20 \\ way & 0.10 \\ figure & 0.02 \\ style & 0.01 \\ & . \\ & . \\ & . \end{bmatrix} \qquad form \begin{bmatrix} forme & 0.57 \\ manière & 0.15 \\ formulaire & 0.10 \\ mode & 0.10 \\ genre & 0.03 \\ & . \\ & . \\ & . \end{bmatrix}$$

Figure 2.2 : Distribution des probabilités lexicales pour le mot français ‘‘$forme$’’ et le mot anglais ‘‘$form$’’.

La figure 2.2 montre la distribution des probabilités lexicales correspondant au mot français ‘‘$forme$’’ ainsi qu’au mot anglais ‘‘$form$’’.

Tout le processus qui permet d’apprendre la distribution des probabilités lexicales est appelé processus ou phase d’**entraînement** des modèles de traduction à base de mots.

2.3 Modèles de traduction à base de segments ou de séquences de mots

Certes, l’entraînement des modèles de traduction à base de mots permet de déterminer la distribution des traductions lexicales et de s’en servir pour générer la traduction de phrases. Cependant, il existe dans la littérature une meilleure approche qui permet d’exploiter les probabilités lexicales ainsi que les alignements déduits des modèles de traduction à base de mots. La famille de modèles ainsi construits sont connus sous le nom de modèles à base de segments ou de séquences de mots « Phrase-Based Models, (**PBM**) ». Ces derniers sont retenus dans la plupart des travaux récents comme les modèles de référence. Notons que dans ce qui suit, nous tenons compte du fait que le sens de la traduction est inversé dans le modèle mathématique. Nous utilisons alors la notation $p(\mathbf{f}|\mathbf{e})$ pour désigner le modèle de traduction.

2.3.1 Phase d’entraînement

Le modèle de traduction $p(\mathbf{f}|\mathbf{e})$ est exprimé en fonction de la probabilité conditionnelle de la traduction particulière d’une séquence de mots $p(\bar{f}|\bar{e})$ au lieu des probabilités de traductions lexicales. $p(\bar{f}|\bar{e})$ est estimée par la fréquence relative :

$$p(\bar{f}|\bar{e}) = \frac{compte(\bar{f},\bar{e})}{\sum_{\bar{f}_l} compte(\bar{f}_l,\bar{e})} \tag{2.9}$$

$\bar{f}$ est une séquence de m mots dans la langue source $(f_p \dots f_{m+p-1})$.

$\bar{e}$ est une séquence de l mots dans la langue cible $(e_q \dots e_{l+q-1})$.

$compte(\bar{f},\bar{e})$ désigne le nombre de fois où la la séquence de mots $\bar{f}$ est traduite par $\bar{e}$.

$\sum_{\bar{f}_i}$ représente la somme sur toutes les séquences de mots de la langue source dont la traduction coïncide avec la séquence de mots $\bar{e}$ de la langue cible.

Pour pouvoir entraîner le modèle de traduction des séquences de mots, nous devons développer une méthode permettant d'extraire les paires de segments de mots $(\bar{f}, \bar{e})$. Les paires qui sont considérées dans la traduction sont celles qui sont consistantes avec un alignement de mots relatif à $(\mathbf{f}, \mathbf{e})$. Une paire de segments de mots $(\bar{f}, \bar{e})$ est consistante avec un alignement A donné si tous les mots de $\bar{f}$ alignés dans A ne sont alignés qu'avec tous les mots de $\bar{e}$ qui possèdent un alignement dans A et vice versa. La figure 1.7 illustre un tel concept où la séquence de mot ''*assumes that*'' est alignée avec la séquence de mots allemands ''*geht davon aus , dass*''.

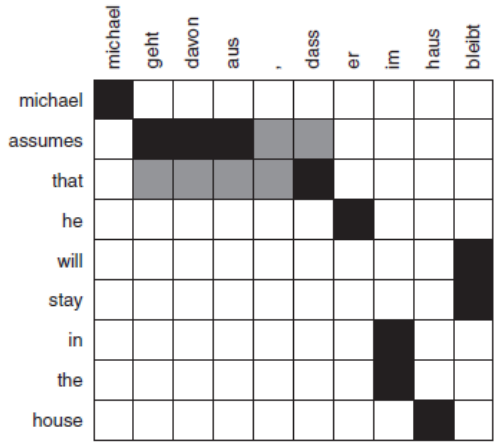

Figure 2.3 : Extraction d'une paire de segments consistante avec l'alignement représenté par la matrice d'alignement (Koehn et Ltd, 2010).

Outre le modèle de traduction des segments de mots $p(\bar{f}_i|\bar{e}_i)$ et le modèle de langue $p(\mathbf{e})$, d'autres éléments peuvent être intégrés dans le modèle de traduction de phrases $p(\mathbf{e}|\mathbf{f})$ pour l'améliorer en intégrant d'autres éléments. Les trois suivants sont typiquement mis à contribution :

1. Un modèle de réordonnancement qui permet de capturer l'arrangement des mots dans la traduction.

2. Une pénalité sur les mots qui permet d'ajuster leurs longueurs, empêchant ainsi le modèle d'avoir une tendance à produire des mots très courts ou très longs.
3. Une pénalité sur les séquences de mots qui permet d'ajuster la longueur des segments d'une manière analogue à la pénalité de mots.

2.3.2 Modèle de traduction log-linéaire

Le modèle de traduction $p(\mathbf{e}|\mathbf{f})$ s'exprime comme une combinaison des éléments précédents. Dans la combinaison, chaque élément est pondéré par un coefficient qui traduit sa contribution au modèle de traduction $p(\mathbf{e}|\mathbf{f})$. Dans le cas simplifié où le modèle de traduction intègre uniquement le modèle de traduction des segments $p(\overline{f_i}|\overline{e_i})$ et le modèle de langue $p(\mathbf{e})$ alors $p(\mathbf{e}|\mathbf{f})$ prend la forme multiplicative suivante :

$$p(\mathbf{e}|\mathbf{f}) = \prod_{i=1}^{I} p(\overline{f_i}|\overline{e_i})^{\alpha_1} . p(\mathbf{e})^{\alpha_2} \quad (2.10)$$

Où I représente le nombre de segments de mots alignés entre **f** et **e.** L'ajout des pondérations nous permet désormais d'exprimer le modèle de traduction $p(\mathbf{e}|\mathbf{f})$ **en fonction** d'un modèle log-linéaire :

$$p(\mathbf{e}|\mathbf{f}) \propto \exp\left[\sum_{i=1}^{I} \alpha_1 \log\left(p(\overline{f_i}|\overline{e_i})\right) + \alpha_2 \log\left(p(\mathbf{e})\right)\right] \quad (2.11)$$

Dans l'équation (2.10), le log de chaque modèle intégré apparaît comme un attribut du modèle log-linéaire.

Plus généralement, lorsque le modèle de traduction $p(\mathbf{e}|\mathbf{f})$ comporte M attributs, l'équation (2.10) prend la forme suivante :

$$p(\mathbf{e}|\mathbf{f}) \propto \exp\left[\sum_{j=1}^{M} \alpha_j \, m_j \, (\mathbf{f}, \mathbf{e})\right] = p_{\alpha_1^M}(\mathbf{e}|\mathbf{f}) \quad (2.12)$$

m_j représente l'attribut associé au $j^{\text{ème}}$ élément intégré dans la combinaison (m_j peut être un modèle de langue, un modèle de réordonnancement, ou n'importe quel autre modèle). α_j désigne le poids qui lui correspond.

2.3.3 Phase de développement (mise au point)

Les valeurs optimales des coefficients de pondération (α_j) sont celles qui donnent le meilleur modèle de traduction (la traduction la plus vraisemblable ou la plus probable). La méthode la plus populaire pour estimer ces valeurs optimales est celle qui est implémentée dans **Moses**. Le critère d'optimisation utilisé dans ce cas est le « Minimum Error Rate Training (**MERT**) » ou taux d'erreur d'apprentissage minimal (Och, 2003). Pour une métrique d'évaluation donnée, par exemple, l'indicateur **BLEU** (voir section 2.4.1), le principe de MERT repose sur la recherche du vecteur de poids $\widehat{\alpha_1^M}$ qui permet d'obtenir le score **BLEU** le plus élevé sur un corpus de textes contenant la référence R pour la traduction ainsi que le texte en langue source F. Ce corpus est appelé ensemble de développement et il contient en général quelques milliers de phrases. Le vecteur $\widehat{\alpha_1^M}$ optimal est alors déterminé comme l'indique l'équation (2.12) :

$$\widehat{\alpha_1^M} = argmax_{\alpha_1^M} \text{BLEU}\left[(argmax_E \log\left(p_{\alpha_1^M}(E|F)\right)), R\right] \tag{2.13}$$

$argmax_{\alpha_1^M}$ maximise le score **BLEU** en fonction du texte de référence R. $argmax_E$ maximise le log du modèle de traduction appliqué au texte F de la langue source pour trouver la meilleure traduction. (Och, 2003) propose une solution à ce problème d'optimisation en utilisant un sous-ensemble des meilleures traductions pour chaque phrase. Ceci est effectué en itérant à chaque fois après mise à jour du vecteur de poids $\widehat{\alpha_1^M}$ selon l'équation (2.13). La nouvelle valeur de $\widehat{\alpha_1^M}$ est utilisée pour ajouter de nouveaux candidats aux listes des meilleures traductions. Ainsi après ajouts des candidats, les listes sont utilisées de nouveau pour mettre à jour $\widehat{\alpha_1^M}$ selon l'équation (2.12). Ce processus itératif est répété jusqu'à ce que les listes des meilleures traductions ne changent plus. Nous référons le lecteur à (Cherry et Foster, 2012) pour une comparaison de plusieurs techniques d'optimisation permettant de résoudre ce problème.

Le processus permettant de déterminer les valeurs optimales des poids est appelé processus ou phase de **développement « Tuning »**.

2.3.4 Phase de décodage

Une fois le modèle de traduction entrainé (appris, estimé), il reste à **décoder** les traductions des phrases à traduire et qui n'ont pas été observées lors des phases d'entraînement et de développement. On rappelle que ces deux phases consistent à faire l'apprentissage des distributions des modèles intégrés dans le modèle de traduction et des coefficients de pondération relatifs aux modèles intégrés. Le décodage, selon le jargon de l'apprentissage statistique, est la phase de test ou de généralisation des modèles construits au cours des phases d'entraînement et de développement. La tâche de décodage qui consiste à trouver la traduction la plus probable aux nouvelles données non observées est difficile. (Knight, 1999) démontre que le problème qui consiste à choisir la meilleure traduction parmi toutes les traductions possibles sur la base d'un score est un problème NP-complet.

Pour éviter ce problème, plusieurs heuristiques ont été proposées. **Moses** utilise une recherche en faisceau qui peut s'énoncer comme suit : La recherche de la meilleure hypothèse de traduction se fait en partant d'une hypothèse vide (aucune traduction n'est encore produite). L'hypothèse vide est par la suite étendue pour donner une arborescence d'hypothèses alternatives qui couvrent plus de mots sources. Un score est affecté à chaque hypothèse traduisant ainsi la probabilité de sa réalisation. Ce score est utilisé pour éliminer de l'espace de recherche les hypothèses les moins prometteuses. La figure1.8 montre un exemple d'une arborescence d'hypothèses.

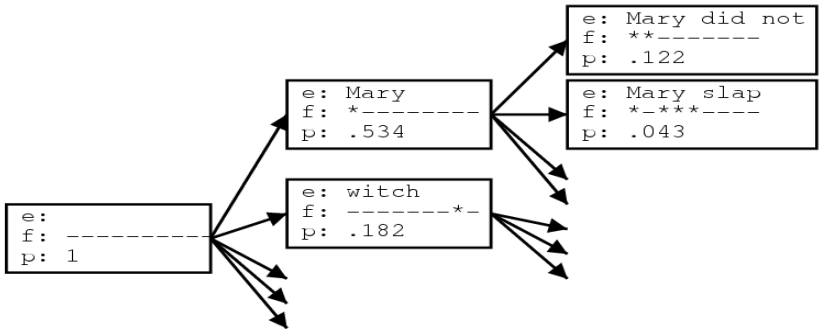

Figure 2.4 : Arborescence des hypothèses de traduction[4]

[4] http://www.statmt.org/moses/?n=Moses.Background

2.4 Critères d'évaluation

2.4.1 La métrique BLEU

Dans cette thèse nous retenons le score **BLEU,** introduit par (Papineni et al., 2002), comme critère principal d'évaluation de la performance prédictive des modèles que nous proposons. La métrique retenue est la plus indiquée pour cette fonction puisqu'elle corrèle hautement avec l'évaluation humaine. Un score **BLEU** est calculé en fonction des n-grammes produits et qui sont retrouvés dans leurs références de traductions respectives. L'équation (1.11) donne l'expression du score.

$$\text{BLEU} = bp\ .\ exp\ (\sum_{i=1}^{n} \omega_i log\ (précision_i)) \tag{2.14}$$

la précision est donnée par :

$$précision_i = \frac{\sum_{t,r} \sum_{Seg_i \in t} min[nb(Seg_i,t),nb(Seg_i,r)]}{\sum_{t,r} \sum_{Seg_i \in t} nb(Seg_i,t)} \tag{2.15}$$

nb désigne une fonction qui permet le calcul de la fréquence d'un segment dans une phrase. Seg_i est un i-grammes, t est une traduction candidate et r est une référence à t. bp est une pénalité qui permet de réduire le score **BLEU** des traductions très courtes. Dans le cas où la taille de la traduction candidate est supérieure à la référence, bp prend la valeur égale à 1. Dans le cas contraire, elle prend la valeur $1 - \frac{taille\ référence}{taille\ traduction}$. En général, tous les ω_i prennent la même valeur. Par exemple dans le cas où $n = 4$, tous les ω_i sont fixés à $\frac{1}{4}$. Dans la pratique les scores **BLEU** sont calculés à l'aide de segments d'au plus quatre mots et donc de 4-grammes. Pour pouvoir intuitivement comprendre cette métrique et comment un jugement **BLEU** est attribué, nous illustrons le tableau 2.1. Le tableau illustre un exemple d'une évaluation **BLEU.** Les n-grammes qui sont présents dans la phrase de référence sont marqués en couleur ou chaque couleur correspond à un ordre n.

2.4.2 Les métriques SER et WER

À côté de la métrique **BLEU**, les systèmes de traduction conçus dans ce mémoire seront évalués par les métriques **WER** et **SER**.

Référence	Can you give me some medicine for headache ?	BLEU
Traduction 1	Can I have some medicine for headache ?	47.75
Traduction 2	Can you give me prescribe some medicine ?	37.71

n-grammes correct où n ≥ 4

2-grammes correct

1-gramme correct

Tableau 2.1: Exemple illustrant des scores **BLEU**[5]

La métrique **WER** (Word Error Rate) est calculée à l'aide de la distance d'édition séparant une traduction candidate de sa référence. La distance d'édition est le nombre minimal d'opérations de suppression, d'insertion ou de remplacement à effectuer pour transformer une chaîne de caractères en une autre (Tillmann et al., 1997). Le score **WER** est obtenu en normalisant la distance d'édition par la taille de la référence, en termes de caractères. Le score **WER** traduit le degré de non-conformité ou (non-authenticité) de la traduction et de la référence. Autrement dit lorsque **WER** augmente la conformité ou la similarité de la traduction à la référence diminue.

La métrique **SER** (Sentence Error Rate) mesure le taux de phrases traduites non identiques aux phrases de référence. Comme pour la métrique **WER,** on note que le degré de similarité ou de conformité de la traduction à la référence varie en sens inverse avec la mesure **SER**.

Les deux indicateurs de sélection et de performance **WER** et **SER** ont la même importance que le score **BLEU**. Néanmoins, **WER** et **SER** ont l'inconvénient et le risque de pouvoir omettre ou éliminer à tort des traductions potentiellement correctes. Ces dernières sont

[5] http://www.iro.umontreal.ca/~felipe/IFT6010-Automne2012/Transp/traduc.pdf

considérées par **WER** et **SER** des traductions erronées, car ces métriques ne tiennent compte que de la ressemblance entre la référence et la traduction candidate.

2.4. Résumé

Dans ce chapitre nous avons passé en revue les propriétés et le fonctionnement des deux grandes familles de modèles statistiques constituant l'une des composants principaux des systèmes de traductions automatiques. Ce sont les modèles à base de mots et baptisés modèles **WBT ou «** Word-Based Translation models **»** et les modèles à base de séquences désignés par **PBT** ou « Phrase-Based Translation models ». La spécification de ces modèles repose essentiellement sur l'alignement, qui n'est pas souvent procuré, de mots ou des séquences de mots. Pour les modèles de la première famille, durant la phase d'entraînement un modèle apprend à calculer la distribution des traductions lexicales qui servent d'entrée pour la génération automatique de la traduction de phrases. Les sorties ainsi générées constituent des entrées pour initier les modèles de la seconde famille. Ces derniers sont retenus dans la plupart des travaux récents comme les modèles de référence (ou benchmark).

Les phases de développement et de décodage sont introduites. Pour l'étape d'évaluation, des métriques indiquant la performance prédictive des systèmes mis en place sont présentées de manière critique quant à leurs points forts et points faibles. Les soubassements théoriques des concepts et des procédures introduites dans ce chapitre sont adoptés dans le reste des chapitres.

Chapitre 3 Calibration d'un système de traduction SMT de base pour la traduction de l'anglais vers l'inuktitut

Dans ce chapitre nous proposons un système de traduction de base de l'anglais, considéré comme une langue à morphologie pauvre, vers l'inuktitut, qui est une langue à morphologie complexe. Ce travail s'insère dans le cadre d'un projet de recherche mené au sein du laboratoire de Recherche Appliquée en Linguistique Informatique (**RALI**). Le choix de l'Inuktitut est motivé par le fait que c'est une langue qui fait réunir toutes les difficultés et les lacunes que peut rencontrer un système de traduction automatique SMT. En effet contrairement aux langues à morphologie pauvre, où l'ordre des mots revêt une importance syntaxique particulière, l'Inuktitut a une morphologie dans laquelle l'ordre des mots dans une phrase n'est pas grammaticalement proéminent. Dans ce dernier cas, les statistiques usuelles calculées à partir des n-grammes deviennent obsolètes, car elles ne sont plus informatives (Clifton, 2010). Par ailleurs, comme pour les langues morphologiquement complexes, les mots de l'Inuktitut sont généralement composés de plusieurs morphèmes. L'agrégation du lexique morphologique peut alors conduire à un problème de rareté des données ou « data sparcity problem ». Un autre aspect des difficultés que peut rencontrer un système de traduction automatique est le problème d'asymétrie de la source et de la cible. En effet pour une langue cible qui est agglutinative, en l'absence d'information morphologique, l'implémentation du modèle de traduction requiert alors l'intégration de composantes plus complexes.

Le choix de l'inuktitut est aussi motivé par le fait que c'est une langue qui n'a pas été étudiée dans le « MT Summit Shared Task » par (Koehn, 2005) et dont on veut connaître la qualité de la traduction qui lui est associée.

Le système de traduction de référence que nous proposons pour effectuer la traduction automatique de l'anglais vers l'Inuktitut est basé sur l'approche « Phrase-Based Translation (**PBT**)». Le reste de ce chapitre est organisé comme suit. La section 3.1 donne une description des données utilisées pour les étapes d'entraînement, de développement et

de test du système de traduction. La section 3.2 précise le protocole expérimental et la phase de prétraitement des données. Les détails relatifs à l'expérimentation et aux composantes du système de traduction retenu sont reportés dans la section 3.3. La section 3.4 apporte une lecture des résultats enregistrés alors que la section 3.5 est réservée à la conclusion de ce chapitre.

3.1 Données utilisées

Notre première expérience consiste à traduire du texte de l'anglais vers une langue morphologiquement riche qui est l'inuktitut. L'inuktitut est la langue des Inuits qui vivent au Nunavut qui se trouve au nord-ouest du Canada. Comme il a été mentionné dans l'introduction, l'inuktitut est une langue très agglutinative, c'est-à-dire qu'un mot en inuktitut est formé d'une collection de morphèmes. En reprenant l'exemple, mentionné dans (Johnson et Martin, 2003), le mot ''*qaisaaliniaqquunngikkaluaqpuq*'' est composé par les morphèmes ''*saali*'', ''*niaq*'', ''*qquu*'', ''*nngit*'', ''*galuaq*'' et ''*puq*'' dont la traduction en anglais donne la phrase ''*Actually he will probably not come early today*''.

Dans la plupart des cas, un mot en Inuktitut est traduit, en anglais, en plusieurs mots. Le grand nombre de morphèmes dans l'exemple mentionné ci-dessus est problématique pour la traduction, car la forme prise par ces morphèmes est variable. Une telle variabilité est illustrée par les suffixes ''*nngit*'', et ''*galuaq*'' qui, combinés ensemble, donnent ''*nngikkaluaq*''. L'exemple précédent illustre le problème de la complexité morphologique de l'inuktitut.

Pour l'entraînement, le développement et le test[6] des modèles de traduction retenus dans notre expérimentation, les Hansards anglais et inuktituts de l'assemblée législative du Nunavut pour la période du 1er avril 1999 jusqu'au 8 novembre 2007, à l'exception de l'année 2003, ont servi comme corpus parallèle. (La phase de test ou d'évaluation est la phase de décodage des nouvelles données qui n'ont pas servi ni à l'entraînement et ni au

[6] http://www.inuktitutcomputing.ca/NunavutHansard/fr/index.html

développement. Les données traduites sont évaluées par rapport à la référence de test). Pour effectuer l'alignement des phrases nous avons fait appel au programme basé sur l'algorithme de (Gale et Church, 1993). Le corpus contient initialement 535000 phrases alignées. Nous avons utilisé pour la tâche d'entraînement et de développement l'outil de traduction **Moses** (Koehn et al., 2007).

3.2 Protocole expérimental et prétraitement des données

Lorsqu'un corpus contient des phrases alignées dont la longueur est hétérogène d'une langue à une autre, cela peut constituer un problème pour le système de traduction. Par ailleurs, l'outil d'alignement **Giza++,** utilisé par **Moses**, consomme énormément de ressources en terme de temps nécessaire à l'alignement des phrases longues.

Pour réduire l'effet négatif d'hétérogénéité sur la qualité de traduction, et dans un souci d'avoir une meilleure allocation des ressources en termes de temps, nous avons procédé à l'élimination des phrases qui appartiennent au corpus et dont la longueur est supérieure à 40. Un prétraitement de tokénisation et de suppression des majuscules a été parallèlement effectué. (La tokénisation est l'identification des unités de mots dans un texte. Elle consiste principalement à séparer la ponctuation des mots). Ce filtrage préliminaire nous a permis d'obtenir un corpus parallèle formé de 506162 phrases alignées. Dans ce corpus les tailles des vocabulaires inuktitut et anglais sont reportées dans le tableau 3.1 :

Langue	**Nombre de mots (ou tokens) total**	**Nombre de mots distincts (Taille du vocabulaire)**
Anglais	5603078	526344
Inuktitut	3153724	30749

Tableau 3.1 : Statistiques relatives aux vocabulaires inuktitut et anglais

Les données consignées dans le tableau 3.1 exhibent la richesse de l'inuktitut. En effet il est facile de constater que le volume du vocabulaire inuktitut est 17 fois plus grand que celui du vocabulaire anglais. Le corpus ainsi obtenu a été primitivement partitionné en trois sous-ensembles. Le premier contient 470000 phrases alignées et sert à l'entraînement des modèles de traduction. Les deux autres comportent, chacun, 1000 phrases destinées aux tâches de développement et de test respectivement. Dans notre protocole expérimental, nous avons choisi de subdiviser les deux ensembles de test et de développement en plusieurs sous-ensembles de phrases alignées. Le choix d'une telle partition est motivé par notre souci d'accélérer la tâche de développement et, surtout, de pouvoir comparer les résultats sur la base de plusieurs fichiers lors de la phase d'évaluation de la performance des modèles de traduction entretenus. Par ailleurs, cette stratégie d'affinage permet de tester aussi la « significativité » des résultats. Pour l'ensemble de développement, la répartition adoptée comprend 4 sous-ensembles dont deux de dimension 200 phrases et deux de dimension 300 phrases. L'ensemble de test a été scindé en 10 sous-ensembles contenant chacun 1000 phrases.

3.3 Expérimentation

Pour la traduction de l'anglais vers l'Inuktitut nous avons utilisé dans notre expérimentation le logiciel de traduction statistique **Moses.** Le système de traduction référence retenu est basé sur l'approche « Phrase-Based Translation (**PBT**)» avec ses trois fonctions usuelles à savoir : un modèle de langue 3-gramme, un modèle de traduction et un modèle de réordonnancement. Pour la spécification et la construction du modèle de langue, nous avons eu recours au logiciel **SRILM** (Stolcke, 2002). La traduction a été accomplie en utilisant des valeurs standards pour les hyperparamètres du système à savoir : une taille maximale de 20 mots pour une séquence, une limite de réordonnancement de 6 mots, une pile d'hypothèses de taille 100 et un maximum de 20 séquences de mots candidates à la traduction d'un segment de mots en langue source. Le tableau 3.1 présente un aperçu des 40 expériences établies.

Fichier Test	Fichier Développement	WER	SER	BLEU
Test 1	Dev 1	45.19	78.20	30.63
Test 2	Dev 2	44.42	77.10	28.45
Test 3	Dev 3	41.24	73.30	33.73
Test 4	Dev 3	41.95	76.00	34.73
Test 5	Dev 4	51.14	88.60	24.81
Test 6	Dev 1	36.94	66.00	40.91
Test 7	Dev 2	46.95	83.70	30.78
Test 8	Dev 3	48.31	84.50	29.97
Test 9	Dev 3	44.25	77.20	33.29
Test 10	Dev 1	49.25	84.70	30.59
Moyennes		44.96	78.93	31.79
Moyennes de toutes les expériences		**45.10**	**79.60**	**31.43**

Tableau 3.2 : Résultats des expériences relatives à la traduction anglais-inuktitut

3.4 Interprétation des résultats

La première constatation pouvant être établie du tableau 3.2 est que la qualité de la traduction produite dépend considérablement du jeu de données d'entraînement et de test utilisé. On observe une différence remarquable entre les scores **BLEU** relatifs aux différents jeux de données, et ce, malgré le fait que les sous-ensembles d'évaluation et les sous-ensembles de développement utilisés font partie du même corpus.

Nous pouvons aussi affirmer, que les résultats obtenus dans le tableau précédent pour les expériences relatives à la traduction de l'anglais vers l'inuktitut ne sont pas, le moins qu'on puisse dire, attendus et même surprenants puisqu'on obtient des performances de traduction exceptionnelles. Ce constat est érigé par des scores **BLEU** élevés, variant

entre 24.81 et 40.91, ainsi que des taux d'erreurs au niveau des mots et des phrases (**WER** et **SER**) réduits. Cela ne devrait pas être le cas étant donné que l'inuktitut est une langue morphologiquement complexe et comporte un vocabulaire très volumineux, ce qui rend la tâche de traduction plus difficile comme l'a indiqué (Koehn, 2005). Nous rappelons que dans « MT Summit Shared Task », (Koehn, 2005) a montré que les systèmes les moins performants sont ceux qui sont relatifs à des tâches de traduction dont la langue cible est morphologiquement riche. Les meilleurs scores obtenus sont relatifs à des tâches de traduction n'incluant pas de langues morphologiquement complexes et les scores **BLEU** varient entre 30 et 40 pour de telles traductions.

Pour avoir une meilleure compréhension des résultats enregistrés par le système de traduction anglais-inuktitut, et afin d'apprécier à leur juste valeur, les modèles de traduction mis en place, nous avons été amenés à procéder au calcul des statistiques relatives aux vocabulaires inuktitut et anglais utilisés au cours des trois phases d'entraînement de développement et de test.

En ce qui concerne le vocabulaire test de l'anglais, ce dernier comporte 5340 mots distincts dont 5053 sont observés dans les phases d'entraînement et de développement. Les 113286 occurrences de mots dans l'ensemble de test sont observées durant le développement et l'entraînement. Tandis que le nombre d'occurrences du vocabulaire non observé est de 337 seulement. Ceci est traduit par le fait que le vocabulaire anglais observé lors des deux phases se produit dans 99.67% des cas dans l'ensemble de test. Ceci pourrait constituer une deuxième explication en ce qui concerne la facilité d'apprentissage du corpus anglais-inuktitut.

Pour plus de rigueur dans l'interprétation des résultats, nous avons envisagé une deuxième procédure d'évaluation qui consiste à mesurer la ressemblance entre l'ensemble de test anglais et les données anglaises qui ont servi à l'entraînement et la configuration (c'est à dire développement) de notre système de traduction. Cette procédure consiste à calculer les distances d'édition les plus proches séparant les données de l'ensemble test et

les données qui ont servi à la configuration (ou développement) et à l'entraînement. Pour pouvoir mesurer cette ressemblance, nous devons aussi établir ce calcul sur les données d'un autre corpus utilisé dans la littérature tel que le corpus anglais-français d'Europarl v6[7]. Ce dernier corpus est aussi une collection de textes parlementaires européens. Les distributions des distances d'éditions les plus proches, séparant les données d'évaluation des données utilisées pour entraîner et configurer le système de traduction pour les deux corpus, sont alors comparées. Nous avons appliqué le même prétraitement au corpus anglais-français que celui relatif aux données inuktitutes. La même stratégie de partition des phrases du corpus anglais-inuktitut a été reconduite pour le corpus anglais-français. Le nombre de phrases relatives à l'entraînement, au développement et au test sont respectivement 470000, 1000 et 10000.

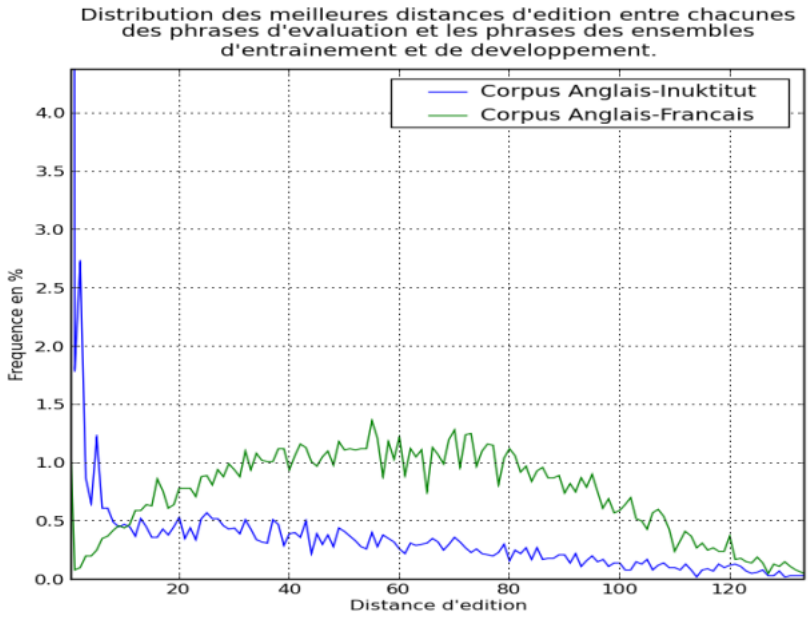

Figure 3.1 : Distributions des distances d'édition

La figure 3.1 illustre les deux distributions relatives aux deux corpus. Pour les données de notre corpus anglais-inuktitut, la courbe est descendante et prend une forme qui

[7] http://www.statmt.org/europarl/v6/

s'amortit. La plupart des distances d'édition sont dans ce cas proches de 0. À l'opposé, pour le corpus anglais français d'Europarl v6, la courbe n'est pas monotone et la plupart des distances d'édition sont comprises entre 20 et 100. Une image plus précise sur ce comportement est obtenue en zoomant sur la partie des distances d'édition proche de 0. L'histogramme de la figure 3.2 indique que plus de 58% des phrases de l'ensemble test ou d'évaluation du corpus Hansards anglais-inuktitut sont observées dans l'ensemble d'entraînement et de développement. Cette ressemblance entre les données d'évaluation et celles qui sont utilisées aux phases d'entraînement et de développement, est due au fait que le Hansards anglais-inuktitut est spécifique au jargon utilisé par les membres de l'assemblé et contient un nombre important de phrases dupliquées. Les phrases les plus fréquentes des données d'entraînement anglaises, comme ''*thank you , mr. speaker*'', ''*thank you , mr. chairman*'' ou encore ''*chairperson (interpretation) :*'', sont retrouvées aussi comme les phrases les plus fréquentes dans l'ensemble d'évaluation. Chacune de ces phrases est produite plus d'une dizaine de milliers de fois dans les données d'entraînement et plus d'une centaine de fois dans les données d'évaluation.

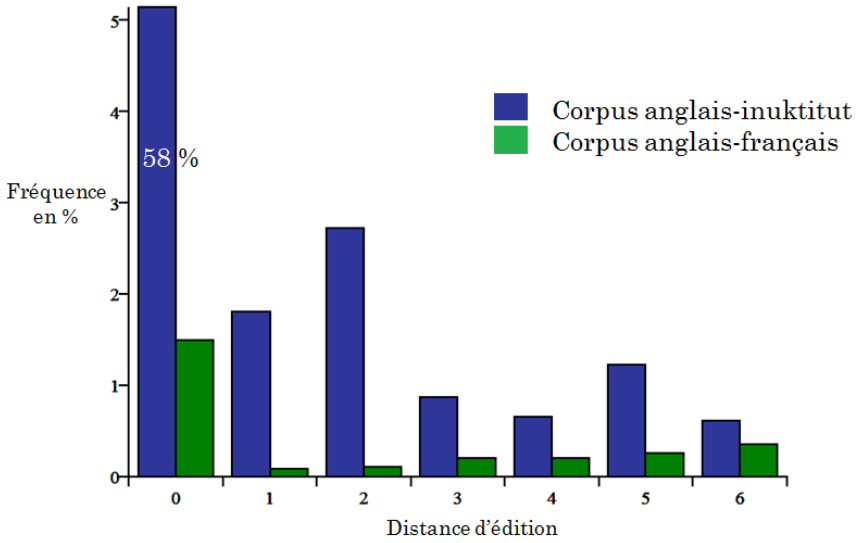

Figure 3.2 : Zoom sur l'histogramme des distances d'édition les plus proches des phrases d'évaluation

Par contre, en ce qui concerne le corpus français-anglais d'Europarl v6, 1.49 % seulement des phrases anglaises appartenant à l'ensemble test sont observées dans l'ensemble d'entraînement. Les résultats expliquent la facilité d'apprentissage du corpus inuktitut-anglais et tous ces faits nous mènent à dire que le corpus anglais-inuktitut est plus facile à apprendre.

3.5 Résumé

Dans ce chapitre, nous avons testé un système de traduction de base dans une tâche de traduction vers une langue à morphologie complexe qui est l'inuktitut. L'expérimentation du système retenu pour la traduction de l'anglais vers l'inuktitut montre une performance prédictive appréciable du système de traduction. La bonne qualité des résultats de traduction, indiquée par les valeurs prises par les métriques usuelles telles que le score **BLEU**, et les taux d'erreurs au niveau des mots et des phrases (**WER** et **SER**), est imputable à la facilité d'apprentissage du corpus inuktitut-anglais dû au fait que les données d'évaluation sont similaires aux données observées lors des phases d'entraînement et de développement.

Une solution permettant d'éviter ce problème de surapprentissage est d'épurer le corpus en enlevant les phrases dupliquées. Ceci permet d'obtenir un jeu de test différent des données de développement et d'entraînement. Cependant, il semble plus habile de travailler avec une paire de langues pour laquelle ce problème de surapprentissage ne se pose pas et pour laquelle on peut se comparer à l'état de l'art. Nous avons alors choisi, dans le cadre du chapitre 4, de travailler avec le finnois comme langue cible et pour lequel les données d'apprentissage et de développement dont on dispose ne sont pas similaires aux données d'évaluation. Par ailleurs, la traduction de l'anglais vers le finnois a suscité un intérêt particulier de la communauté scientifique. Cet intérêt est conforté par la disponibilité d'un grand nombre de travaux qui seront utilisés dans un exercice de « benchmarking ». Notons aussi que nous essaierons de tester une combinaison d'un prétraitement et d'un post-

traitement pour la tâche de traduction anglais-inuktitut pour vérifier si les résultats obtenus pour la tâche de traduction anglais-finnois concordent avec celle de l'inukitut.

Chapitre 4 Combinaisons de techniques de prétraitement et de post-traitement de base pour la capture de l'information morphologique : étude de cas sur le finnois et sur l'inuktitut

Dans ce chapitre nous proposons plusieurs systèmes de traduction statistique pour la tâche de traduction d'une langue à morphologie pauvre vers une langue à morphologie complexe. Pour pallier aux insuffisances causées par la complexité morphologique et afin d'améliorer la qualité de la traduction, nous présentons des systèmes de traduction avec des traitements de données à priori (prétraitement) et à postériori (post-traitement). Nous appliquons ce type de transformations pour traduire du texte anglais vers du finnois. La particularité morphologique du finnois indique la nécessité de procéder à de telles opérations. Nous appliquons des techniques de prétraitement relativement simples et d'autres, plus développées. Les techniques de prétraitement simples consistent à couper chaque mot en une unité lexicale dont la taille est au plus égale à k caractères. Le but d'utiliser de telles techniques et de pouvoir comparer leur efficacité à capturer de l'information morphologique par rapport aux techniques les plus développées telles que la segmentation non supervisée ou la stemmatisation à l'aide d'un algorithme tenant compte des caractéristiques morphologique de la langue en question. En outre, contrairement au chapitre précédent les données d'évaluation ne sont pas similaires aux données observées lors des phases d'entraînement et de développement. Le reste du chapitre est organisé de la manière suivante. La section 4.1 décrit la particularité morphologique du finnois. La section 4.2 retrace le cadre conceptuel général du processus de traduction qu'on propose ainsi que les différents types de prétraitement (stemmatisation, segmentation non supervisée ou combinaison des deux) et de post-traitement (désambigüisation ou prédiction morphologique, accolage de segments). La section 4.3 est réservée à la partie expérimentale et à l'interprétation des résultats.

4.1 La particularité Morphologique du finnois

Le finnois fait partie des langues finno-ougriennes qui sont parlées en Hongrie, aux pays baltes, en Finlande, en Russie et en Scandinavie. Le finnois est une langue morphologiquement riche. Cette richesse est illustrée par la production des formes fléchies qui peuvent être dérivées à partir d'un stemme, appelé aussi thème morphologique. Un thème morphologique est la partie du mot ne contenant pas son suffixe. Par exemple, en français, les mots ‘‘*établissons*’’ et ‘‘*établissez*’’ correspondent respectivement aux suffixes ‘‘–s*ons*’’ et ‘‘*–sez*’’ et partagent le même stemme ‘‘*établis*’’. Pour mieux comprendre les notions citées, nous illustrons des exemples mentionnés dans (Moscoso del Prado Martín, Bertram, Häikiö, Schreuder, et Baayen, 2004). Un stemme peut être associé à des milliers de formes fléchies. Par exemple, en finnois, le stemme ‘‘*työ*’’, qui veut dire travail, est associé à 7000 formes fléchies. Parmi ces formes on trouve : ‘‘*työntekijä*’’ qui veut dire employé, ‘‘*työläs*’’ qui signifie laborieux, ‘‘*työehtosopimus*’’ qui veut dire traité du taux de salaire, ‘‘*työväenluokka*’’ qui désigne la classe ouvrière, etc. Même si la plupart des stemmes finnois sont associés à un nombre réduit de formes fléchies, un grand nombre de stemmes peut donner lieu à plusieurs centaines de ces formes.

4.2 Processus de traduction proposé

Comme le finnois est une langue morphologiquement riche, un stemme quelconque peut être associé à plusieurs formes fléchies. L'agrégation du lexique morphologique et du stemme peut alors conduire à une explosion combinatoire de formes. Ce nombre explosif de formes fait diminuer les chances ou les fréquences de leurs occurrences dans le texte et crée un problème de rareté des données ou « data sparcity problem ». Il s'ensuit que pour un texte morphologiquement complexe le nombre de mots en dehors du vocabulaire aurait tendance à augmenter rendant ainsi les statistiques relatives à l'occurrence des mots de moins en moins robustes.

Pour réduire l'effet de cette complexité morphologique, nous proposons dans ce chapitre de concevoir des systèmes de traduction avec des traitements de données à priori

(prétraitement) et des traitements à postériori (post-traitement). Nous appliquons ce type de transformations pour traduire du texte anglais vers du finnois. Un seul type de ces systèmes sera appliqué dans la tâche de traduction de l'anglais vers l'inuktitut. Il est important de signaler que les opérations de prétraitement et de post-traitement des données sont ici optionnelles. On peut, par exemple, générer le finnois à l'aide d'un système de traduction établissant des alignements entre les mots anglais et les mots finnois sans que ces derniers ne soient prétraités. Par contre, l'opération de post-traitement des données ne peut avoir lieu que s'il y a eu, au préalable (avant la traduction), un prétraitement des données.

La figure 4.1 retrace le cadre général du processus qu'on propose pour la traduction de l'anglais vers le finnois. L'apprentissage du système de traduction est tout d'abord accompli à l'aide des données d'entraînement qui peuvent être prétraitées. Les données de développement servent quant à elles à la configuration et à l'optimisation des paramètres du modèle candidat à la traduction. Bien évidemment, les données de développement doivent être prétraitées dans le cas où les données d'entraînement le sont. Une fois que ces tâches sont accomplies, on peut alors générer des sorties en finnois (prétraité) à partir des données anglaises appartenant à l'ensemble d'évaluation. Les données ainsi générées peuvent être traitées à postériori s'il y a lieu.

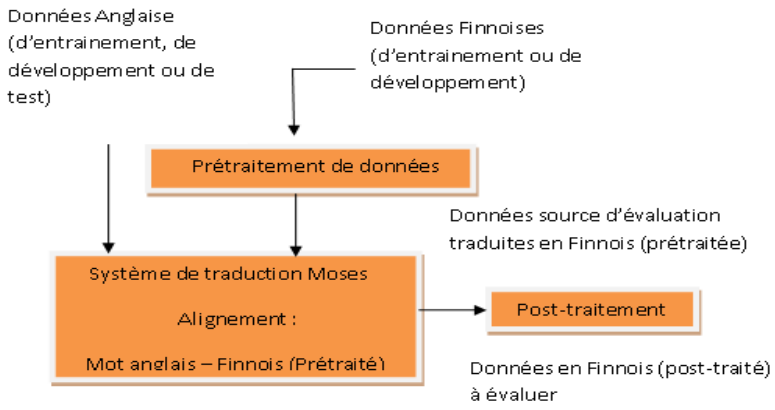

Figure 4.1 : Processus général de la Traduction

4.2.1 Opérations de prétraitement

Dans les expériences qui vont être décrites plus tard, nous nous sommes servis de deux types de prétraitements. Le plus simple est appelé *stemming* ou parfois stemmatisation. Le stemming est l'opération qui consiste à réduire un mot à son stemme, ou thème morphologique. Cela peut s'effectuer d'une manière arbitraire ou à l'aide d'un algorithme tenant compte des caractéristiques morphologiques de la langue concernée. Un tel algorithme permettrait de conserver la partie d'un mot ne contenant pas son suffixe. (Watanabe, Tsukada, et Isozaki, 2006) suggèrent une façon arbitraire pour effectuer le stemming. Il s'agit de couper chaque mot en une unité lexicale dont la taille est au plus égale à k caractères.

Le deuxième type de prétraitements pouvant être appliqué aux données finnoises est la segmentation. En effet, la segmentation consiste à diviser le mot en des segments correspondants à des morphèmes. On rappelle qu'un morphème est défini comme étant la plus petite unité lexicale portant un sens (Matthews, 1991). Une telle opération permet de considérer les morphèmes obtenus comme étant les unités lexicales de la traduction. Les résultats de segmentation peuvent servir comme entrée à la démarche de stemmatisation. Dans certains cas, la segmentation peut jouer le rôle d'un outil de stemming. Dans ce cas, le dernier segment de chaque mot segmenté sera considéré comme son suffixe et sera alors supprimé. La segmentation peut être réalisée à l'aide d'outils tenant compte des caractéristiques morphologiques de la langue en question. La segmentation peut aussi être accomplie à l'aide de techniques d'apprentissage non supervisé. Ces techniques permettent d'apprendre la structure morphologique des mots d'une manière optimale, c'est-à-dire, de construire un vocabulaire de morphèmes réduit à partir duquel, on peut générer n'importe quel mot en concaténant les segments faisant partie du vocabulaire. La plupart des travaux portant sur la traduction automatique montrent que l'utilisation de la segmentation non supervisée est plus efficace que la l'utilisation de la segmentation supervisée (Clifton et Sarkar, 2011).

Les différents types de prétraitements, exposés plus haut, requièrent des transformations à postériori (post-traitement) permettant de rétablir le finnois généré par le système de traduction en finnois correct.

4.2.2 Opérations de post-traitement

Comme il a été énoncé auparavant, la phase de post-traitement ne peut avoir lieu que lorsque la condition d'un prétraitement est remplie. En fait, le post-traitement est une opération qui sert à transformer le finnois prétraité, généré par le système de traduction, en finnois correct. Il existe plusieurs types de post-traitements. Ceux-ci dépendent, notamment, de la nature du prétraitement appliqué aux données d'entrée. En outre à chaque type de prétraitement correspond un type de post-traitement particulier.

Si la tâche de stemmatisation est exécutée au cours de la phase de prétraitement, une tâche de désambigüisation (ou de prédiction) morphologique devrait alors se être établie à la phase de post-traitement. La désambigüisation consiste à convertir un texte d'un vocabulaire V1 en un texte d'un vocabulaire V2. Dans notre cas, il s'agit de prédire les suffixes des mots stemmés (ou stémmatisés). Par ailleurs, un post-traitement d'accolage des segments devrait être élaboré dans le cas où la segmentation a servi comme tâche de prétraitement des données. Dans le cas où la segmentation est utilisée comme outil de stemming, une désambigüisation morphologique des données devrait être établie, mais l'accolage des segments est réalisé cette fois à la phase de prétraitement (voir ultérieurement figure 4.4 section 4.3.1).

Pour tous les traitements mentionnés ici, il existe un certain nombre d'outils qui facilitent leurs implémentations sur les données en question. Le principe de fonctionnement de ces outils est expliqué dans la section qui suit.

4.2.3 Outils de prétraitement

4.2.3.1 Outils de Stemming

Comme mentionné dans 2.2.1, le stemming des mots finnois et inuktituts peut être réalisé d'une manière arbitraire, en coupant chaque mot en une unité lexicale dont la taille est au plus égale à *k* caractères. Par exemple, dans le cas où *k*=7, le mot ''*television*'' serait réduit au stemme ''*televis*'' Dans le système de traduction proposé on s'est contenté d'appliquer ce type de stemming uniquement sur les mots qui ne contiennent que des lettres. Comme autre alternative au stemming, nous avons aussi utilisé **Snowball stemmer** (Porter, 2001) pour la langue finnoise. **Snowball** permet d'éliminer la partie suffixe des mots en se basant sur les caractéristiques morphologiques de la langue en question. De tels outils sont malheureusement rares et même inexistants pour certaines langues négligées comme l'inuktitut. Comme outils de prétraitement, seul le stemming arbitraire sera appliqué dans la tâche de traduction de l'anglais vers l'inuktitut.

En général, pour les langues indo-européennes, comme l'italien, le français et l'anglais, et les langues ouraliennes, comme le finnois, le stemming est un outil largement adopté et appliqué[8]. Pour ces langues, les suffixes peuvent être répartis en 3 classes baptisées le *i-*, le *d-* et le *a-suffix.*

Un *a-suffix*, ou enclitique est un mot attaché à un autre mot. Comme par exemple, en français, les enclitiques ''*puis-**je***'' ou ''*allons-**y***''. En italien, les pronoms personnels attachés à certains verbes, comme ''*mandar**gli***'' qui veut dire ''envoyer + lui'', sont considérés comme des enclitiques.

Le *i-suffix*, ou suffixe flexionnel, constitue une marque du genre, du temps ou de la personne[9]. Par exemple le ''***esse***'' dans ''*duch**esse***'' qui est le féminin de *duc*. En anglais, le passé des verbes est formé en ajoutant le ''***ed***'' comme dans ''*start**ed***''. Un *d-suffix*, ou suffixe dérivationnel, sert à former de nouveaux mots à partir des radicaux[8]. En anglais, le

[8] http://snowball.tartarus.org/texts/introduction.html
[9] http://www.cnrtl.fr/definition/suffixe

‘‘***ness***’’ qui peut être ajouté à certains adjectifs pour former les noms qui leur correspondent, est un suffixe dérivationnel, comme par exemple, ‘‘*kind**ness***’’. Les nouveaux mots formés peuvent avoir des catégories grammaticales ou des sens différents du mot radical. En français, le ‘‘***ette***’’**,** comme dans ‘‘*maisonn**ette***’’**,** modifie le contenu sémantique du radical puisqu'il permet de particulariser la petitesse de la maison. En général, les suffixes dérivationnels sont suivis par les suffixes flexionnels qui eux, sont suivis par les enclitiques. On devrait donc supprimer dans l'ordre les enclitiques, puis les suffixes flexionnels et enfin les suffixes dérivationnels. Les distinctions entre les enclitiques, les suffixes flexionnels et les suffixes dérivationnels peuvent être établies en finnois. Dans **Snowball**, on veut éliminer tous les *a-*, les *i- suffixes* et quelques suffixes dérivationnels. Le système de terminaison s'applique aux nominaux à savoir les noms, les adjectifs et les pronoms. Pour donner une idée des sorties générées par les procédures de stemming, on mentionne l'exemple illustré par le tableau 4.1.

Type de stemming	**Exemple**
Sans stemming	*julistan perjantaina joulukuun 17 .*
Snowball	*julist perjant jouluku 17 .*
Réduction des mots en k caractères (k=7)	*julista perjant jouluku 17 .*

Tableau 4.1 : Exemples de phrases finnoises stémmatisées

4.2.3.2 Outils de segmentation

Pour accomplir cette tâche nous utilisons les données qui nous ont été gracieusement fournies par (Clifton et Sarkar, 2011). Les informations reçues de ces auteurs contiennent les analyses morphologiques correspondant à leurs données d'entraînement, de développement et d'évaluation. Ces analyses morphologiques ont été effectuées à l'aide de l'outil de segmentation non supervisée **Morfessor** (Creutz et Lagus, 2005). Dans le Morpho Challenge 2008 (Kurimo, Turunen, et Varjokallio, 2009), le résultat généré par l'équipe gagnante a été établi en combinant les résultats générés par **Morfessor** (Creutz et Lagus, 2005) et **Paramor** (Monson, 2008). (Monson, Carbonell, Lavie, et Levin, 2009) expliquent comment les analyses morphologiques de **Morfessor** et de **Paramor** ont

été combinés pour la compétition. Ces derniers ont soummis séparément les analyses morphologiques produites par **Paramor** et **Morfessor** pour chaque mot, produisant ainsi une segmentation ambigüe entre **Paramor** et **Morfessor** pour chaque mot. Le défi était de concevoir et d'évaluer des algorithmes d'apprentissage non supervisé permettant d'établir une analyse morphologique des mots en différentes langues. Pour le finnois, le turque, l'arabe et l'allemand, les meilleures performances ont été obtenues par le système combinant **Morfessor** et **Paramor**. Dans une des compétitions du Morpho Challenge 2010 (Kurimo, Virpioja, et Turunen, 2010), la tâche consistait à évaluer les algorithmes d'analyse morphologique dans un contexte de traduction statistique. L'évaluation concerne la sortie générée par le système de traduction pouvant traduire de l'allemand ou du finnois analysé vers l'anglais. En ce qui concerne la traduction de l'allemand vers l'anglais, on obtient le meilleur résultat en terme de score **BLEU** en combinant les sorties respectives du modèle de traduction utilisant **Morfessor** (Creutz et Lagus, 2005) et du modèle de traduction standard (sans analyse morphologique). Quant à la traduction de l'anglais vers le finnois, le meilleur résultat obtenu en termes de score **BLEU** est relatif à deux systèmes de traduction utilisant respectivement **Morfessor Baseline** et **Morfessor Categories-MAP**, qui sont deux variantes de **Morfessor**. Le choix de (Clifton et Sarkar, 2011) s'est porté ici sur **Morfessor Categories-MAP** pour accomplir la tâche de segmentation des données.

Morfessor produit, dans une première étape, une segmentation initiale des mots en morphes. Un morphe étant la réalisation d'un morphème dans un contexte particulier. Ceci est réalisé en utilisant la méthode récursive de description de la longueur minimale « Minimun Description Length (**MDL**) » , décrit dans le rapport technique de (Creutz et Lagus, 2005). Un vocabulaire ou lexique de morphes est construit d'une manière qui permet de représenter n'importe quel mot du corpus à partir d'une concaténation de morphes. Le but est alors de trouver la segmentation et le lexique optimaux. L'algorithme **MDL** parcourt tous les mots du corpus. Parmi les segmentations possibles qu'un mot pourrait avoir, c'est la segmentation la plus probable qui est finalement retenue. Dans cette étape, les morphes sont choisies selon leur fréquences dans le corpus et aucune catégorie morphologique (préfix, stemme, suffixe) n'est utilisée. Les vraisemblances des mots sont

modélisés par des modèles de Markov cachés « Hidden Markov Models (**HMM**) » ayant une seule catégorie permettant d'émettre le morphe. Ce processus est récursivement répété jusqu'à ce que les probabilités de segmentation convergent. En procédant ainsi on aboutit souvent à une sur-segmentation des mots rares représentés par la concaténation des segments fréquemment présents dans le lexique. Une autre conséquence de l'emploi d'une telle méthode, est la sous-segmentation des mots fréquemment présents dans le corpus qui dans la plupart des cas, restent non segmentés. Ce comportement découle du fait que la représentation optimale du corpus est obtenue quand un mot fréquemment présent est entièrement représenté dans le lexique.

Dans une seconde étape, un modèle probabiliste de maximum a posteriori (MAP) est utilisé pour analyser la segmentation établie à la première étape. Ceci est réalisé en créant une représentation hiérarchique des morphes. Un morphe, lors de cette étape, peut être représenté par 2 sous-morphes, lesquels, peuvent être aussi représentées récursivement par des sous-morphes et ainsi de suite. Un morphe peut ne pas avoir de sous-morphes et dans ce cas il est le représentant de lui même. La figure 3 illustre une représentation hiérarchique d'un mot finnois décomposé en plusieurs morphes.

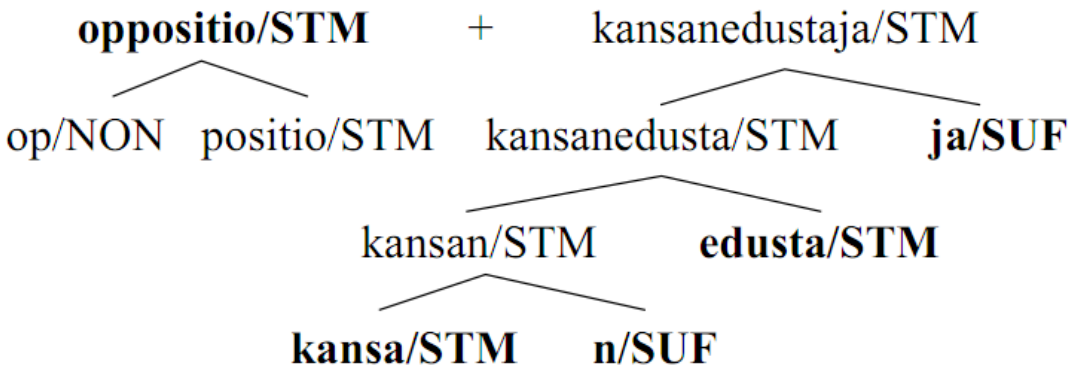

Figure 4.2 : Représentation hiérarchique d'un mot finnois décomposé en plusieurs morphes (Creutz et Lagus, 2005)

ici, le mot finnois ''*oppositionkansanedustaja*'' n'est pas fréquemment présent dans le corpus. Cependant, ''*oppositio*'' qui veut dire opposition et ''*kansanedustaja*'' qui veut dire membre du parlement, le sont. La segmentation donne ''*oppositio*+ +*kansa*+ +*n*+

+*edusta*+ +*ja*''. Les distributions de probabilités des mots sont représentées par des modèles de Markov cachés. Les états cachés représentent les catégories morphologiques (préfix, stemme, suffixe, non-morphème). Ces catégories émettent des morphes (ou des segments) avec des probabilités particulières. Une transition d'un préfixe vers un suffixe n'est permise que si une transition intermédiaire émettant un stemme est effectuée. Comme dans la première étape, le but dans cette étape est de trouver la segmentation et le lexique optimaux. L'estimateur de maximum a posteriori (**MAP**) est alors exprimé par la formule suivante :

$$argmax_{lexique}\, p(lexique|corpus) = argmax_{lexique}\, P(corpus|lexique)\,.\,P(lexique) \quad (4.1)$$

Le pseudo-code suivant énumère les étapes établies par **Morfessor** permettant d'effectuer la segmentation :

1. Initialiser la segmentation.
2. Fractionner les morphes.
3. Accoler les morphes en utilisant une stratégie ascendante.
4. Fractionner les morphes.
5. Resegmenter le corpus en utilisant l'algorithme de Viterbi et ré-estimer les probabilités jusqu'à ce qu'elles convergent.
6. Répéter les étapes 3,4 et 5.
7. Étendre la représentation hiérarchique du morphe à la résolution la plus fine ne contant pas de non-morphèmes.

L'initialisation de l'algorithme de segmentation est obtenue par le biais d'une recherche effectuée par l'algorithme **MDL**. Les segments établis par cette initialisation sont alors étiquetés par des catégories morphologiques. À partir de l'étape suivante, c'est le modèle **MAP**, comme mentionné et formulé précédemment, qui se charge de l'analyse de la segmentation obtenue. Au cours de la deuxième et la quatrième étape, qui consistent à fractionner les morphes, toutes les divisions possibles d'un morphe en sous morphes sont testées et la plus probable est alors gardée. La troisième étape consiste à joindre les

morphes ensembles afin d'en former de plus longs. La structure morphologique ainsi que la concaténation optimale sont alors retenues.

4.2.4 Outils de post-traitement

4.2.4.1 Procédure d'accolage des segments

En adoptant la segmentation comme prétraitement, le système de traduction est alors entraîné en ayant comme unité lexicale les segments de mots dont les stemmes et les suffixes sont séparés par des ''+ +''. Pour avoir une meilleure idée de l'entrée finnoise, illustrons l'exemple d'une phrase anglaise et de sa traduction segmentée. La traduction de la phrase anglaise ''*thank you , mr segni , i shall do so gladly* .'' est la phrase finnoise suivante ''*kiitos , jäs+ +en seg+ +ni , teen sen oike+ +in mielellä+ +ni .''* Une fois que le système produit la sortie générée, une procédure d'accolage des stemmes et des suffixes doit être établie. Cet accolage consiste à recoller les segments séparés par ''+ +'' pour former les mots finnois. L'accolage de la phrase finnoise précédente donnerait ''*kiitos jäsen segni , teen sen oikein mielelläni* .'' Cependant, une fois que la traduction est générée, la procédure d'accolage n'est pas évidente puisque des cas particuliers, ne respectant pas les règles d'affichage des segments, peuvent être produits. L'accolage doit tenir compte de ces cas. Par exemple, une sortie de la forme ''seg1+ seg2'' peut être générée. Illustrons l'exemple d'un cas particulier qui a été généré par le système de traduction relatif au modèle de segmentation. La traduction produite de la phrase anglaise ''*and even the european politicians have insufficient insight into the negotiating process*'' est la suivante ''*ja jopa euroop+ +an poliitik+ +ot ovat riittämät+ +tömiä* ***kuv+ neuvotteluproses****+ +s*i''. Pour palier aux insuffisances relevées dans la procédure d'accolage, nous avons utilisé la commande linux **sed** qui permet de remplacer les occurrences d'une chaîne de caractères par une autre. Nous remplaçons tout d'abord tous les ''+ +'' par la chaîne vide. Puis nous remplaçons tous les ''+ '' et les '' +'' par la chaîne vide et enfin nous éliminons tous les ''+'' restants en les remplaçant par la chaîne vide. Avec une telle méthode, la sortie du processus de traduction donnerait : ''*ja jopa euroopan poliitikot ovat riittämättömiä*

kuvneuvotteluprosessi .'' La simplicité d'une telle procédure d'accolage est justifiée par le fait que le modèle de traduction de phrases $p(\mathbf{e}|\mathbf{f})$ tient compte de traitements plus complexes. Par exemple**,** un modèle de réordonnancement est intégré au modèle de traduction de phrases $p(\mathbf{e}|\mathbf{f})$ lors de l'entraînement afin de capturer l'arrangement des unités atomiques (ici les segments) dans la traduction**.**

4.2.4.2 Outils de génération morphologique

La procédure de stemming permet de supprimer la partie suffixe des mots. Si une telle procédure est implémentée à la phase de prétraitement, une prédiction morphologique doit être établie. Cette prédiction permet de générer la partie manquante des mots produits par le système de traduction. La génération peut être réalisée à l'aide de l'outil de désambigüisation **Disambig** de **SRILM**[10] de (Stolcke, 2002). **SRILM** est un étiqueteur **HMM** qui traduit une séquence de mots dans un vocabulaire V1 en une séquence de mots faisant partie d'un vocabulaire V2. La désambigüisation est opérée selon un « mapping » reliant un stemme à plusieurs formes fléchies. Le « mapping » permet de joindre un stemme à une forme fléchie par une probabilité $p(forme\ fléchie|stemme)$ qui représente la fréquence relative où la forme fléchie a été associée au stemme en question dans le corpus. Pour un stemme donné, un tel « mapping » est exprimé par une table contenant des observations ou instances. Une instance d'une telle table est transcrite de la manière suivante :

$stemme\ forme\_fléchie1\ p(forme\_fléchie1|stemme)$

$forme\_fléchie2\ p(forme\_fléchie2|stemme)$

...

L'exemple suivant illustre cette représentation pour le stemme ''*televisio*'' :

''televisio televisio**n** 0.333333 televisio**ista** 0.033333 televisio**on** 0.033333 televisio**ta** 0.033333 televisio 0.100000 televisio**sta** 0.166667 televisio**ssa** 0.300000''

L'ambigüité dans cet exemple est résolue en trouvant la séquence de mots du vocabulaire

[10] http://www.speech.sri.com/projects/srilm/manpages/disambig.1.html

V2, correspondant aux formes de surfaces, étant donné la séquence de mots du vocabulaire V1 des stemmes, $p(\mathrm{V2}|\mathrm{V1})$. Cette probabilité est calculée à partir de la probabilité conditionnelle $p(\mathrm{V1}|\mathrm{V2})$ et d'un modèle de langue pour les séquences de mots du vocabulaire V2. Dans notre cas, on utilisera des modèles de langues 3-grammes.

$$argmax_{V2}p(\mathrm{V2}|\mathrm{V1}) = argmax_{V2}p(\mathrm{V1}|\mathrm{V2})p(\mathrm{V2}) \quad (4.2)$$

4.3 Expériences

Pour des raisons de comparabilité et de conformité avec les systèmes de traduction anglais-finnois reportés dans les travaux de recherche récents, nous avons choisi d'utiliser le corpus Europarl v3. En effet, ce corpus est considéré comme le corpus de référence dans la plupart des travaux portant sur les modèles statistiques de traduction automatique **SMT** (Luong et al., 2010) et (Clifton et Sarkar, 2011). (Clifton et Sarkar, 2011) nous ont parallèlement fourni les données qu'ils ont utilisées pour entraîner, développer et évaluer les différents systèmes de traduction qu'ils ont conçu.

Dans toutes les expériences que nous avons mené, nous nous sommes servis du logiciel **Moses** (Koehn et al., 2007) pour concevoir et expérimenter des systèmes de traduction basés sur l'approche de traduction à base de séquences d'unités lexicales (**PBT**).

Les différentes procédures de prétraitement et de post-traitement présentées dans les sous-sections 4.2.3 et 4.2.4, peuvent être appliquées aux données afin de générer la traduction finnoise. Dans le cas où le prétraitement s'avère utile, ce dernier est appliqué aux données finnoises d'entraînement et de développement. Rappelons que le prétraitement peut s'effectuer par le biais d'une opération de stemming ou bien de segmentation.

4.3.1 Description des expériences relatives aux opérations de stemming

Comme déjà mentionné, le stemming peut être employé en réduisant la taille de chaque mot (token ou unité lexicale) en une unité dont la taille maximale est égale à k caractères. L'algorithme **Snowball** peut être aussi considéré comme une alternative au

stemming pour les mots finnois. Lorsque le stemming est employé au niveau de la phase de prétraitement, un post-traitement devrait être appliqué aux données finnoises générées en format prétraité. Les données finnoises sont générées par le système de traduction permettant d'aligner les mots anglais avec les unités prétraitées (stemmes). La figure 4.3 décrit les processus de traduction relatifs aux procédures de stemming.

(Clifton et Sarkar, 2011) se sont servi de la segmentation proposée par **Morfessor** pour établir un type de stemming particulier. Dans cette opération particulière chaque token du corpus est décomposé en plusieurs segments. Le dernier segment de chaque token est alors supprimé. Cette segmentation permet de retenir l'information morphologique du mot en question tout en réduisant la taille du vocabulaire des stemmes. La segmentation du mot ''*käytettävä*'', par exemple, donne ''*käy*+ +*t*+ +*et*+ +*tä*+ +*vä*''. Le suffixe ''*vä*'' est alors supprimé. . Comme entrées finnoises au système de traduction, les caractères ''+'' reliant les segments d'un stemme sont supprimés. Cette suppression facilite l'interprétation de la première sortie du système en la considérant comme étant un stemme complexe (stemme et morphe accolé).

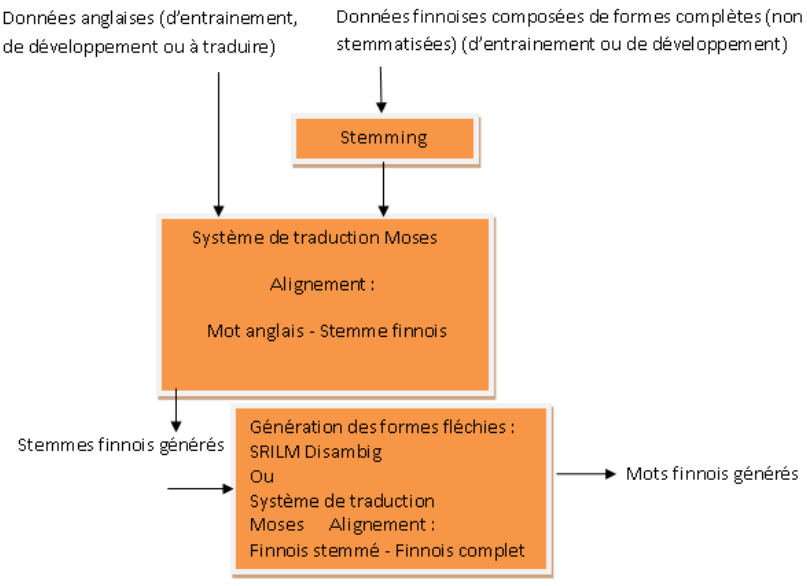

Figure 4.3 : Processus de traduction en stemmes et génération des formes finnoises complètes.

Les données ainsi stemmées nous ont été procurées par (Clifton et Sarkar, 2011). Une fois le finnois stemmé est généré, les auteurs emploient les modèles de champs aléatoires conditionnels ou Conditionnel Random Fields (**CRF)** pour générer les suffixes manquants.

Pour notre système de traduction, la prédiction morphologique des suffixes est établie en utilisant **SRILM Disambig** de (Stolcke, 2002)**.**

En ce qui concerne ce processus, la seule différence entre celui établie par (Clifton et Sarkar, 2011) et le notre, c'est que ces derniers ont entraîné les CRF en établissant des classes d'équivalence pour les voyelles finnoises faisant partie des suffixes. Par exemple, les suffixes *-kö* et *-ko* prennent la forme générique *-kO*. Ceci est dû au fait que l'utilisation des CRF nécessite un nombre limité de sorties. Cette procédure permet de réduire la taille du vocabulaire relatif aux suffixes finnois. Après la génération des suffixes par les CRF, (Clifton et Sarkar, 2011) utilisent un modèle de langue bigramme pour prédire les voyelles. Dans notre cas, on n'établit aucune classe d'équivalence en ce qui concerne les voyelles finnoises vu que nous n'avons pas de contrainte sur le nombre de sortie distinctes. La figure 4.4 décrit un tel processus de traduction.

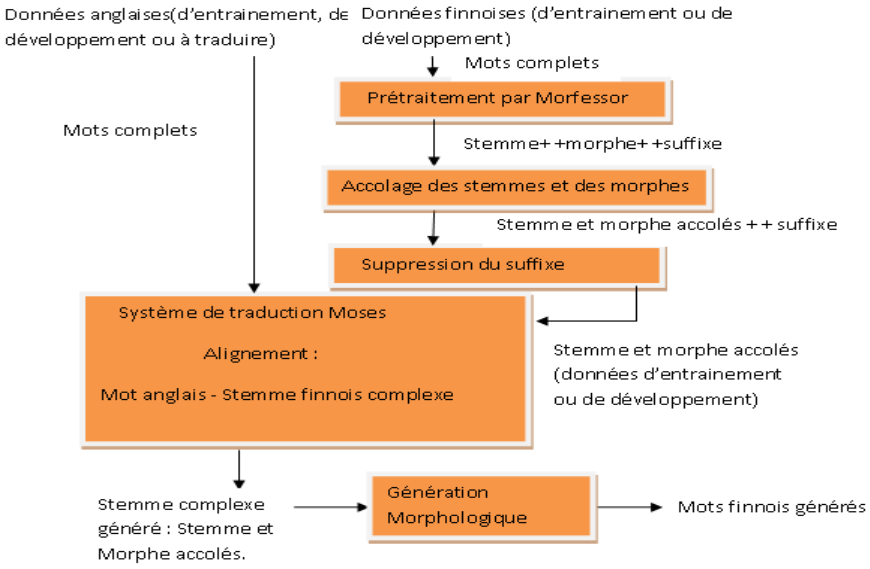

Figure 4.4 : Processus de traduction relatif à la génération morphologique du finnois

Pour avoir une meilleure idée sur le déroulement des phases de prétraitement et de post-traitement décrits ci-dessus, nous illustrons quelques exemples relatifs à ces traitements dans les tableaux 4.2, 4.3 et 4.4.

4.3.2 Description des expériences relatives à l'opération de segmentation

Nous décrivons tout d'abord, la segmentation proposée par (Clifton et Sarkar, 2011). Ces derniers ont utilisé **Morfessor** pour entraîner leur « segmenteur » sauf que la configuration relative à cette procédure est différente de la précédente. Cette segmentation coupe le mot en deux segments, c'est-à-dire, en un stemme et un suffixe. La segmentation du mot ‘‘*käytettävä*’’ donne, par exemple ‘‘*käy+ +tettävä*’’.

Le même type de segmentation a été appliqué aux 5000 mots les plus fréquents de l'ensemble qui a servi à l'entraînement du modèle de traduction de référence (sans prétraitement). Une telle opération sous-segmente le corpus puisque rares sont les mots qui sont segmentés. La phrase finnoise suivante illustre un tel phénomène : ‘‘*polttoaineen hintakysymyskin vaikuttaa minusta erityisen tärkeä+ +ltä viime ai+ +kojen tapahtumien valossa.*’’ (Clifton, 2010).

Afin d'enrichir le vocabulaire des segments et éviter le problème posé par la sous-segmentation, les mots non segmentés, mais qui contenaient les suffixes obtenus à partir de la première segmentation ont été aussi segmentés. Cette segmentation supplémentaire est faite de telle manière que le suffixe soit le plus long possible. L'objectif de la recherche du suffixe le plus long est d'abord de construire un modèle de traduction, avec plusieurs exemples de stemmes, ensuite d'enrichir le vocabulaire des suffixes en tenant compte de ceux qui sont rarement présents dans le corpus. Ceci est dû au fait que le modèle ait tendance à sous apprendre ces formes rares vu que la pénalité de mot sanctionne les mots longs.

La procédure décrite, au paragraphe précédent, permet d'obtenir un corpus sur-segmenté, où des formes de mots qui ne devraient pas être segmentées le deviennent.

Procédure de stemming	**Phrase finnoise**
Phrase finnoise avant prétraitement	*kehotan , että nousette seisomaan tämän minuutin hiljaisuuden ajaksi*
Stemming à 7 caractères	*kehotan , että nousett seisoma tämän minuuti hiljais ajaksi*
Snowball	*kehot , et nouset seisom tämä minuut hiljaisuud aja*
Stemming à l'aide de **Morfessor**	*kehotan , että nousette seisomaan tämä minuut hiljaisuuden ajaksi*

Tableau 4.2: Application de différents stemming sur une phrase finnoise

Procédure de stemming	**Traduction produite**
Phrase anglaise à traduire	*they too now have a clear idea of the rights which they have to respect*
Référence finnoise	*heilläkin on nyt selvä käsitys oikeuksista , joiden mukaisesti heidän pitää toimia*
Traduction vers du finnois à 7 caractères	*heillä on nyt selkeä käsity heidän oikeuks kunnioi*
Traduction vers du finniois stemmé par **Snowball**	*ne on nyt selk ajatus oikeuks , jotk niide on kunnioitettav*
Traduction vers du finnois stemmé à l'aide de **Morfessor**	*ne on nyt selke käsitys oikeuksia , joita niiden on noudatettava*

Tableau 4.3 : Traductions produites relatives aux différentes procédures de stemming

Procédure de stemming	**Phrases finnoises produites après désambigüisation morphologique**
Finnois à 7 caractères	*Heillä on nyt selkeä käsitys heidän oikeuksiaan kunnioitetaan*
Finnois stemmé par **Snowball**	*ne on nyt selkeä ajatus oikeuksia , jotka niiden on kunnioitettava*
Finnois stemmé à l'aide de **Morfessor**	*ne on nyt selkeä käsitys oikeuksia , joita niiden on noudatettava*

Tableau 4.4 : Désambigüisations morphologiques relatives aux différentes procédures de stemming

Ainsi, le nombre de segments du côté de la cible devient beaucoup plus grand que le nombre de mots du côté de la source. Pour éviter une telle sur-segmentation, (Clifton et Sarkar, 2011) ont employé des heuristiques dont les détails sont expliqués dans la thèse de (Clifton, 2010). Ces heuristiques consistent à retenir uniquement les suffixes dont la longueur minimale est supérieure à deux caractères et de ne segmenter que les mots dont le stemme a une longueur minimale de deux caractères.

Suite à notre requête, les données d'entraînement, de développement et d'évaluation, ainsi prétraitées, nous ont été procurées par (Clifton et Sarkar, 2011).

Utilisant ces données, nous avons développé un système de traduction permettant de traduire l'anglais vers le finnois segmenté. La procédure d'accolage des stemmes et des suffixes sert alors comme post-traitement des données générées par le système de traduction que nous proposons. Les étapes de tout le processus de traduction sont représentées dans la figure 4.5.

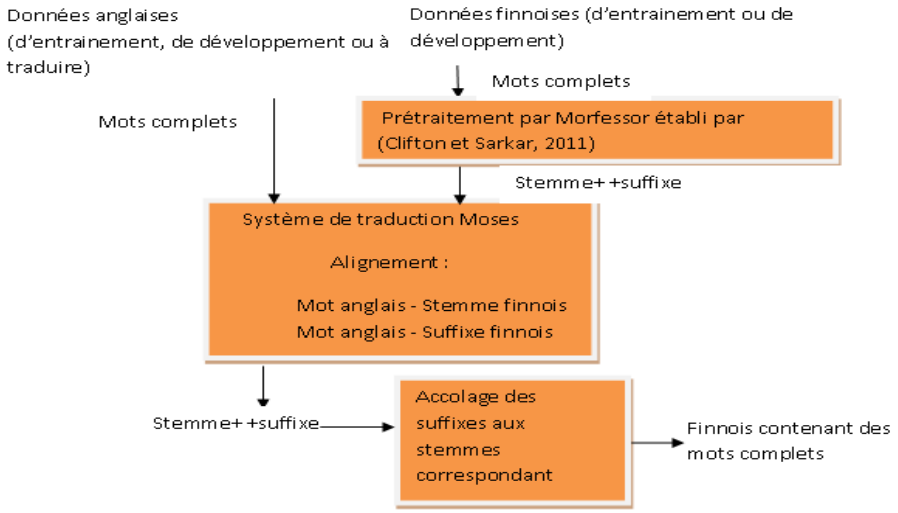

Figure 4.5 : Système de traduction anglais-finnois construit à partir de données segmentées par (Clifton et Sarkar, 2011)

Pour avoir une meilleure idée sur le déroulement de la segmentation et de l'accolage des segments, nous illustrons quelques exemples relatifs à ces traitements :

Phrase finnoise avant segmentation	**Segmentation**
kehotan , että nousette seisomaan tämän minuutin hiljaisuuden ajaksi	*kehot+ +an , että nouse+ +tte seiso+ +maan täm+ +än minuut+ +in hiljais+ +uuden aja+ +ksi*
Phrase anglaise à traduire	*they too now have a clear idea of the rights which they have to respect*
Référence finnoise	*heilläkin on nyt selvä käsitys oikeuksista , joiden mukaisesti heidän pitää toimia*
Traduction Produite	*myös hei+ +llä on nyt selvä käsit+ +ys oike+ +uksia , joi+ +ta nii+ +den on noudat+ +ettava*
Phrase traduite après accolage des segments	*myös heillä on nyt selvä käsitys oikeuksia , joita niiden on noudatettava*

Tableau 4.5: Application de la segmentation, traduction d'une phrase anglaise et accolage des segments produits

4.3.3 Description des données finnoises

Pour chaque processus de traduction, (Clifton et Sarkar, 2011) ont réservé 949924 phrases alignées pour l'entraînement, 2000 phrases alignées pour le développent et 2000 autres pour l'évaluation. Comme il a été déjà indiqué, les corpus prétraités fournis par (Clifton et Sarkar, 2011) sont relatifs aux processus retracés dans les figures 4.4 et 4.5. (Clifton et Sarkar, 2011) nous ont aussi permis d'exploiter le corpus non prétraité.

Même si les corpus procurés contiennent le même nombre de phrases alignées, un examen simple des données révèle une différence marquée entre les phrases de ces corpus. Cette différence dans la composition des corpus n'a pas été signalée par les auteurs. Pour chercher une explication nous avons dû recourir au calcul du nombre maximal de tokens par phrase pour tous les corpus dont nous disposons. Les calculs montrent que pour chacun des corpus, le nombre maximal de tokens par phrase est le même et il est égal à 40. Nous concluons alors que cette différence est vraisemblablement le résultat du prétraitement standard consistant à éliminer les phrases contenant un nombre de tokens supérieur à 40. En effet lorsque les phrases comportant un grand nombre de tokens sont segmentées, celles-ci engendrent un grand nombre de segments qui peut être supérieur à 40. Par contre le nombre de mots lui, peut être inférieur à ce seuil. Ces phrases du corpus sont alors conservées pour le corpus non prétraité alors qu'elles sont éliminées des corpus contenant les données segmentées. Nous avons alors déduit que (Clifton et Sarkar, 2011) ont remplacé ces phrases dans les corpus segmentés par d'autres, dont la taille en termes de segments est inférieure à 40. Ceci a été réalisé dans le but d'avoir un même nombre de phrases pour tous les corpus ce qui explique leur différence observée.

4.3.4 Analyse des données et résultats

Les hyperparamètres et les fonctions standards sont utilisés pour la construction des systèmes de traduction relatifs aux processus proposés et décrits précédemment. (Clifton et Sarkar, 2011) utilisent des modèles de langues 5-grammes. De notre côté, en ce qui concerne le finnois, nous avons construit deux systèmes de traduction relatifs à des modèles

de langues 3-grammes et 5-grammes pour chacun des processus décrits. En ce qui concerne l'inuktitut, nous utilisons des modèles de langues 5-grammes pour le processus de traduction utilisant le stemming arbitraire comme prétraitement et la désambiguïsation par **SRILM Disambig** comme post-traitement**.** Nous avons aussi essayé d'entretenir des modèles de langues 3-grammes, 5-grammes et 7-grammes pour la génération morphologique accomplie par **SRILM Disambig**. Nous avons constaté que cela ne conduit pas forcément à une amélioration importante de la désambigüisation. Le mapping relatif à **SRILM Disambig** a été réalisé avec les données d'entraînement finnoises correspondant au processus de traduction retenu. Le stemming arbitraire (coupant les mots en des unités de *k* caractères) et l'algorithme **Snowball** sont opérés sur les données non prétraitées. Pour les données relatives au stemming réalisé à l'aide de **Morfessor** et décrit dans la figure 4.4, la création du mapping a été obtenue à partir des données qui ont servi à l'entraînement des CRF par (Clifton et Sarkar, 2011). Ces données nous ont été aussi pourvues par ces derniers. Elles contiennent 168906 stemmes distincts.

Pour évaluer l'impact de ces procédures de stemming sur le corpus finnois d'entraînement non prétraité, nous avons appliqué l'algorithme **Snowball** et la réduction des tailles des mots finnois en des stemmes de *k* lettres au plus, sur les données non prétraitées. Les statistiques (Nombre de mots distincts, formes fléchies par stemme) reportées dans le tableau 4.6 illustrent l'amplitude de l'impact recherché.

Opération de stemming	**Nombre de mots distincts**	**Formes fléchies par Stemme**
À 3 caractères	13793	32.77
À 4 caractères	25407	17.25
À 5 caractères	41923	10.45
À 6 caractères	66649	6.57
À 7 caractères	96933	4.52
À 8 caractères	130838	3.35
Snowball	196416	2.23
À l'aide de **Morfessor** (figure 4.4)	168906	1.06

Tableau 4.6 : Formes fléchies et Tailles des vocabulaires relatifs aux stemmings.

Le tableau 4.6 montre que lorsqu'on augmente la valeur du *k,* qui représente la taille des stemmes finnois, la taille du vocabulaire augmente. Ce résultat est attendu puisque lorsque la valeur de *k* augmente, les stemmes et les mots tendent à se confondre. Ceci se confirme en calculant, pour chaque procédure de stemming, la moyenne de formes fléchies associées à un stemme donné. La dernière colonne montre que lorsqu'on augmente la valeur de *k* la moyenne des formes fléchies associée à un stemme donné diminue. Il est cependant utile de noter que le stemming obtenu par l'algorithme **Snowball** produit le vocabulaire d'entraînement le plus grand, mais pas la moyenne de formes fléchies la plus réduite. Malgré le fait que les données d'entraînement stemmées à l'aide de **Morfessor** contiennent un vocabulaire plus petit que les données d'entraînement stemmées par **Snowball,** la moyenne des formes fléchies associées aux données stemmées par **Morfessor** est inférieure à la moyenne des formes fléchies associées aux données stemmées par **Snowball.** Ce résultat montre que l'opération de désambigüisation relative au processus de traduction associé aux données stemmées par **Morfessor** est plus facile. Le tableau 4.7 retrace les résultats des expériences relatives aux différents types de prétraitements et de post-traitements ainsi que les performances des modèles de traduction de langue proposés. Les valeurs affichées par les indicateurs de performance dans le tableau 4.7 montrent la supériorité des modèles de traduction de langue 5-grammes.

Par ailleurs, les différentes procédures de stemming en l'occurrence, l'algorithme **Snowball** et la réduction de la taille des mots en une taille fixe, donnent des scores **BLEU** très médiocres. Une baisse du score **BLEU** est aussi observée pour le tableau 4.9 relatifs aux systèmes de traduction utilisant le stemming arbitraire à 9 caractères pour les mots inuktituts comme prétraitement. On constate aussi que lorsque le découpage des mots finnois devient de moins en moins fin, la performance du système s'améliore de plus en plus. On peut déduire alors, que la réduction du vocabulaire par les procédures de stemming ne permet pas de conserver l'information morphologique des mots. Ceci rend la tâche de désambigüisation plus difficile vu que le choix des formes fléchies devient plus complexe. Ceci explique les scores **BLEU** réduits et les **WER** élevés pour les tâches de traduction

anglais-finnois et anglais-inuktitut. Les expériences menées montrent aussi l'importance du rôle que peut jouer la segmentation en tant qu'outil de prétraitement qui permet d'améliorer la qualité de traduction. En effet les meilleurs résultats enregistrés dans le tableau 4.7 correspondent aux processus de traduction qui utilisent la segmentation comme outil préalable pour transformer les données avant de procéder à la traduction.

On peut donc confirmer que la segmentation permet de retenir l'information morphologique tout en réduisant la taille du vocabulaire et que le stemming ne permet pas de conserver l'information morphologique des mots.

La génération morphologique, à l'aide de **SRILM Disambig,** donne les meilleurs résultats lorsque les données sont segmentées. En termes de score **BLEU**, le système de traduction relatif à la prédiction morphologique des données segmentées produit la meilleure traduction avec un score de 14.80. En termes de **WER**, le processus de traduction basé sur l'approche « **Segmented Translation** » et décrit par la figure 4.5 produit le meilleur résultat.

Malgré les bons résultats enregistrés avec des données segmentées, on ne peut pas confirmer d'une manière certaine et définitive la supériorité de la segmentation en tant qu'outil de prétraitement.

En outre, même si les données d'évaluation sont les mêmes pour tous les processus de traduction, nous ne pouvons pas encore affirmer que les approches basées sur la segmentation sont plus performantes que l'approche de référence (l'approche **Phrase-Based** sans recours à un prétraitement). Ceci est simplement dû au fait que les corpus utilisés pour **entraîner** les systèmes de traduction ne sont pas similaires (ne contiennent pas les mêmes phrases). Pour pouvoir comparer les performances des différents systèmes de traduction et tirer une conclusion de manière intrinsèque, nous devons restituer les mots du corpus dont les unités sont segmentées en deux segments. Ceci peut être assuré en accolant les segments. Cette opération d'accolage permet de concevoir un système de traduction sans aucun Prétraitement.

La procédure de post-traitement (accolage) a été appliquée pour les données relatives à l'approche « **Segmented Translation** ». Celles-ci ont servi à l'apprentissage du processus de traduction décrit dans la figure 4.5. Les résultats obtenus sont consignés dans le tableau 4.9.

Sachant que les données d'évaluation sont les mêmes pour tous les processus de traduction, il n'est pas difficile de constater, d'après le tableau 4.9, que le système de traduction référence (aucun prétraitement) est le plus efficace, enregistrant le score **BLEU** le plus élevé (14.97) et le **WER** le plus petit (63.36) parmi toutes les expériences. La différence entre le système de référence du tableau 4.9 et celui du tableau 4.7 est que celui du tableau 4.9 est entraîné à l'aide des données du corpus utilisé par (Clifton et Sarkar, 2011) pour le système de traduction utilisant la procédure de segmentation par **Morfessor** comme outil de prétraitement.

Ce résultat est en parfaite harmonie avec celui proclamé par (Luong et al., 2010). Les auteurs affirment que certes, le fait de considérer les morphèmes comme unités atomiques de la traduction permet d'améliorer la qualité de l'alignement, cependant cela n'est pas suffisant pour améliorer la traduction. Des méthodes permettant de respecter la forme des mots doivent être aussi réalisées à tous les stades du processus de traduction. Ceci permet de générer comme forme de surface, des mots et non pas des morphèmes incomplets. Nos résultats et ceux de (Luong et al., 2010) ne vont pas de pair avec ceux établis par (Clifton et Sarkar, 2011). En effet ces derniers proclament qu'une segmentation réalisée en ne tenant compte que de l'information monolingue de la langue à segmenter, permet à elle seule d'améliorer la qualité de la traduction. Il s'avère que ceci n'est pas du à une différence de traitement morphologique ou de segmentation mais plutôt au fait que (Clifton et Sarkar, 2011) ont comparé les différents systèmes de traductions en les entraînant sur des corpus contenant des données (phrases) différentes, c'est à dire sur des corpus différents.

		3-grammes			5-grammes		
Prétraitement	**Post-traitement**	**WER**	**SER**	**BLEU**	**WER**	**SER**	**BLEU**
Aucun	Aucun	**64.45**	**97.60**	**14.12**	**64.62**	**97.60**	**14.18**
Stemming à 3 caractères	**SRILM Disambig**	68.92	98.00	10.71	68.85	98.05	11.07
Stemming à 4 caractères	**SRILM Disambig**	66.88	97.95	12.07	66.99	98.05	12.10
Stemming à 5 caractères	**SRILM Disambig**	66.07	98.15	12.47	66.03	98.05	12.55
Stemming à 6 caractères	**SRILM Disambig**	65.63	9795	12.88	65.68	98.00	12.96
Stemming à 7 caractères	**SRILM Disambig**	65.12	97.80	13.26	65.20	97.75	13.19
Stemming à 8 caractères	**SRILM Disambig**	65.19	97.80	13.30	64.84	97.85	13.50
Snowball Stemmer	**SRILM Disambig**	65.49	97.65	12.97	65.32	97.65	13.35
Stemming à l'aide de **Morfessor** (figure 4.5)	**SRILM Disambig**	**63.63**	**97.45**	**14.50**	**63.48**	**97.35**	**14.80**
Segmentation (figure 4.4)	Accolage	**63.71**	**97.65**	**14.26**	**63.33**	**97.35**	**14.71**

Tableau 4.7 : Résultats des expériences de prétraitements et de post-traitements et performance des modèles de langues proposés pour la traduction anglais-finnois.

Prétraitement	Post-traitement	WER	SER	BLEU
Aucun	Aucun	45.10	79.60	31.43
Stemming à 9 caractères	**SRILM Disambig**	45.32	81.09	30.71

Tableau 4.8 : Résultats relatifs aux systèmes de traduction anglais-inuktitut

WER	SER	BLEU
63.36	97.40	14.97

Tableau 4.9 : Résultats relatifs au système de traduction anglais-finnois de référence utilisant les données restituées du corpus segmenté fourni par (Clifton et Sarkar, 2011).

4.4 Résumé étendu

Dans ce chapitre nous avons présenté, dans un premier temps, les soubassements théoriques des systèmes statistiques de traduction automatique susceptibles de capturer les spécificités des structures morphologiques et contourner les difficultés posées par la complexité morphologique. L'architecture de ces systèmes comporte, entre autres, deux composantes de traitement des données à priori (ou prétraitement) (voir détails dans les sections 4.2.2) et à postériori (ou post-traitement) (voir section 4.3.1 et figure 4.4). Le processus de traduction proposé (voir les détails dans les figures 4.1 et 4.3) est appliqué pour la traduction de l'anglais vers le finnois. Cependant, nous avons quand même, testé une expérience pour la tâche de traduction de l'anglais vers l'inuktitut impliquant le stemming comme outil de prétraitement pour vérifier si les résultats obtenus pour la tâche de traduction anglais-finnois concordent avec celle qui est relative à l'inukitut. Nous avons appliqué des techniques de prétraitement relativement simples et d'autres, plus développées. Les techniques simples consistent à couper chaque mot en une unité lexicale dont la taille est au plus égale à *k* caractères. Le but d'utiliser de telles techniques et de pouvoir comparer

leur efficacité à capturer de l'information morphologique par rapport aux techniques les plus développées telles que la segmentation non supervisée à l'aide de **Morfessor** et **Snowball**. Les opérations de prétraitement suggérées (stemming, segmentation, etc.) permettent d'apprendre la structure morphologique des mots, c'est-à-dire, de construire un vocabulaire réduit à partir duquel on peut générer n'importe quel mot. Les transformations antérieures requièrent des opérations de conversions à postériori (accolage des segments, ou désambigüisation, etc.) qui leur sont, dans la plupart des cas, intimement liées. Ces reconversions permettent de rétablir le finnois généré par les systèmes de traduction en finnois correct. Les étapes requises par l'analyse morphologique (section 4.2.3 et 4.2.4) sont décrites pour tous les outils et les algorithmes qui ont servi à l'exécution des opérations de prétraitement et post-traitement des données.

Le deuxième volet couvert dans ce chapitre revêt un aspect pratique consistant à la validation expérimentale des systèmes de traduction proposés. Pour des raisons de comparabilité et de concorde avec les systèmes de traduction anglais-finnois, présentés dans les travaux de recherche récents, notre choix s'est porté sur le corpus Europal v3 qui est considéré comme le corpus référence. Par ailleurs, les étapes d'entraînement, de développement, et d'évaluation, des systèmes retenus, sont effectuées sur les données qui nous ont été procurées par (Clifton et Sarkar, 2011). Ces mêmes données ont servi aux auteurs pour le développement de leurs propres systèmes. La description des données, du protocole expérimental et des expériences de stemming, de segmentation et de désambigüisation sont décrites dans le moindre détail pour tous les systèmes de traductions proposées (section 4.3 et figures 4.3, 4.4 et 4.5).

Les systèmes de traduction retenus dans cette étude expérimentale correspondent à la spécification des modèles de langues à 3-grammes et 5-grammes. Les résultats des expériences relatives aux différents types de prétraitements et de post-traitements ainsi qu'à la performance des modèles de langue sont portés dans les tableaux 4.7, 4.8 et 4.9. Le premier retrace l'impact des opérations de stemming sur le corpus des données non prétraitées. On remarque que lorsque la taille des stemmes finnois augmente la taille du

vocabulaire du corpus (nombre de mots distincts) augmente. Ce résultat est confirmé par la tendance à la baisse de la moyenne des formes fléchies associées au stemme en question. L'algorithme de stemming **Snowball** parvient à produire le vocabulaire d'entraînement le plus grand, mais n'arrive pas à réaliser la moyenne des formes fléchies la plus petite. Le résultat inverse est affiché pour la procédure de stemming à l'aide de **Morfessor** qui permet de générer un vocabulaire plus petit que celui de **Snowball** mais donne une moyenne des formes fléchies supérieure à l'approche de stemming à l'aide de **Morfessor.** Ce résultat montre que l'opération de désambigüisation appliquée aux données stemmées par **Morfessor** est plus facile que celle appliquée aux données stemmés par **Snowball.**

La lecture des résultats du tableau 4.7 révèle la supériorité du modèle de langues 5-grammes. On constate, par ailleurs, que lorsque le découpage des mots finnois, dans une opération de stemming, devient de en moins fin, la performance des modèles de langues est renforcée en conséquence. De là, il n'est pas difficile de conclure que la réduction du vocabulaire par les opérations de stemming n'assure pas la conservation de l'information morphologique des mots finnois. Dans ce cas d'espèce, le choix des formes fléchies devient de plus en plus compliqué, rendant ainsi l'opération de désambigüisation plus difficile à accomplir. En fait, les meilleurs résultats (meilleurs scores **BLEU, SER,** et **WER**) sont inscrits pour les processus de traduction qui utilisent la segmentation comme outil préalable de prétraitement. En effet, la meilleure génération morphologique à l'aide de l'algorithme **SRILM Disambig** est assurée lorsque les données sont segmentées. Le score **BLEU** obtenu dans ce cas est de 14.80. Dès lors, la segmentation peut apparaitre, à la fois, comme un outil de réduction de la taille du vocabulaire et de conservation de l'information morphologique.

Malgré les résultats enregistrés en présence de données segmentées, la supériorité de la segmentation en tant qu'outil de prétraitement ne peut être garantie à priori. Ceci est d'autant plus vrai que les corpus utilisés pour l'apprentissage des systèmes de traduction ne sont pas souvent similaires (ne contiennent pas les mêmes phrases). Les comparaisons entre systèmes de traduction, en présence de données segmentées, ne peuvent se faire, de manière

intrinsèque, qu'après avoir restitué les mots du corpus dont les unités ont été préalablement segmentées. Cette opération est assurée par des transformations d'accolage des segments. L'application de ces transformations aux données relatives au corpus, antérieurement segmenté par (Clifton et Sarkar, 2011), ont permis de restituer les mots de ce corpus. Ces données ont servi à l'apprentissage du processus de traduction référence. Les scores **BLEU**=14.97**,** et **WER**=63.36 enregistrés par ce système référence montrent sa supériorité. Ce résultat concorde avec celui établi par (Luong et al., 2010) et qui stipule que le fait de considérer les morphèmes comme unités atomiques de traduction permet d'améliorer la qualité de l'alignement des mots, mais ceci est insuffisant pour améliorer la traduction. Des méthodes permettant la conservation de la forme des mots doivent, donc, être développées et appliquées à tous les stades du processus de traduction. Nos résultats ne concordent pas avec ceux de (Clifton et Sarkar, 2011) qui affirment qu'une segmentation utilisant l'information morphologique monolingue de la langue permet à elle seule d'améliorer la traduction. Il s'avère que le résultat stipulé par (Clifton et Sarkar, 2011) est du au fait que le système de traduction de référence n'est pas entraîné à l'aide du même corpus utilisé pour entraîner le système de traduction utilisant l'outil de segmentation monolingue **Morfessor** comme outil de prétraitement.

Chapitre 5 Notre approche de Segmentation

Dans le chapitre 4, nous avons mené plusieurs expériences avec différents types de prétraitements et post-traitements et nous avons constaté que la segmentation établie par **Morfessor** ne permet pas à elle seule d'améliorer la qualité de la traduction. (Luong et al., 2010) affirment que si **Morfessor** est employé, la qualité de la traduction ne peut être améliorée que si des méthodes permettant d'assurer, tout au long du processus de traduction, la génération des formes de surfaces de mots, au lieu des morphèmes incomplets, soient employées. Des améliorations de la qualité de la traduction dans des tâches incluant une langue à morphologie riche ont été recensées par (Chung et Gildea, 2009). Ces derniers ne tiennent pas compte seulement de l'information monolingue provenant de la langue à segmenter, mais plutôt de l'information bilingue provenant des deux langues incluses dans la tâche de traduction. Ceci est réalisé en utilisant l'information provenant de la sortie du modèle **IBM 1**.

Notre approche est similaire en philosophie, à celles (Chung et Gildea, 2009), et (Nguyen et al., 2010), mais diffère, de façon significative, quant aux algorithmes de segmentations proposés et à la nature même du schéma de combinaison de ces algorithmes avec des modèles statistiques de traduction automatique à base de séquences d'unités lexicales **PBT**. (Nguyen et al., 2010) utilisent un outil de désambigüisation morphologique propre à la langue à segmenter et la qualité de la traduction n'est pas considérablement améliorée. Dans une tâche de traduction de l'anglais vers le coréen, (Chung et Gildea, 2009) propose une méthode variationnelle bayésienne assez compliquée pour segmenter le coréen. Une amélioration de la qualité de la traduction est observée. Pour accomplir la tâche de segmentation, nous proposons un nouvel algorithme simple, où la segmentation des mots du vocabulaire de la langue cible est établie à partir des valeurs des probabilités de traduction de mots estimées par **Moses**. Cet algorithme de segmentation est facilement généralisable à d'autres langues. Nous nous basons sur l'approche à base de séquences d'unités lexicales **PBT** implémentée dans **Moses** pour la conception de tous les systèmes de traductions testés ici.

Moses génère la distribution des traductions lexicales en utilisant comme information la sortie du modèle **IBM 4**, établi par **Giza++**, et en employant l'heuristique de symétrisation des alignements de mots, expliquée dans l'introduction et conçue par (Och et Ney, 2000). Nous ferons usage de cette distribution comme information bilingue pour la segmentation du vocabulaire finnois. Le reste du chapitre est organisé comme suit. La section 5.1 définit le cadre conceptuel de segmentation. La section 5.2 résume le principe de notre approche et les détails théoriques du paradigme de segmentation. Le pseudo-code de l'algorithme de segmentation est donné en annexe. L'évaluation de la performance prédictive de l'algorithme de segmentation sur un échantillon de 10000 mots finnois est l'objet de la section 5.3. Les procédures de segmentation du corpus finnois sont développées dans la section 5.4. L'évaluation de l'effet des procédures de segmentation proposées sur la qualité de traduction du finnois est reportée à la section 5.5. La section 5.6 clôt le chapitre.

5.1 Cadre conceptuel proposé pour la segmentation

Toutes les segmentations sont réalisées à partir de la distribution des probabilités de traductions lexicales qui est générée par **Moses**. Cette distribution est obtenue à partir des données du corpus non prétraité fourni par (Clifton et Sarkar, 2011). La distribution en question est générée dans un fichier nommé par **Moses** lex.e2f. Le fichier lex.e2f à partir duquel nous concevons notre segmentation, contient les mots anglais ainsi que leurs traductions potentielles en finnois. Pour un mot finnois f quelconque donné, il existe plusieurs mots candidats (potentiels) à sa traduction. On note $P(e|f)$ la probabilité conditionnelle que e soit la traduction de f . Pour chaque mot finnois f la condition d'une distribution de probabilité $\sum_e P(e|f) = 1$ est supposée être remplie. Le tableau 5.1 illustre la distribution des traductions potentielles du mot finnois ''*tieotokoneongelma*''. Le principe général de l'algorithme de segmentation qu'on propose consiste à segmenter les mots finnois étant données leurs traductions anglaises. La tâche consiste à trouver le nombre de segments ainsi que la segmentation optimale étant données les traductions possibles du mot finnois et de ses segments. De cette manière, on voit que l'approche qu'on

Traduction potentielle	Probabilité
Problem	0.1428571
Be	0.1428571
The	0.0952381
By	0.0952381
Affected	0.0952381
With	0.0476190
Will	0.0476190
Was	0.0476190
NULL[11]	0.0476190
Millennium	0.0476190
Could	0.0476190
Computer	0.0476190
Associated	0.0476190
2000	0.0476190

Tableau 5.1 : Distribution de probabilité des traductions potentielles du mot finnois ‘‘*tieotokoneongelma*’’

a adopté permet de capturer et de tirer profit de l’information morphologique bilingue. Par exemple, si le mot finnois ‘‘*tieotokoneongelma*’’ devrait être traduit par ‘‘*computer*

[11] La valeur NULL d’une traduction représente l’évènement où un mot n’est pas traduit

problem'', le mot anglais ''*computer*'' correspondrait au segment finnois ''*tieotokone*'' et le mot anglais ''*problem*'' correspondrait au segment finnois ''*ongelma*''. Comme la combinaison optimale de traductions est relatives aux segments ''*tieotokone''* et ''*ongelma*'', le mot finnois devrait alors être segmenté ainsi en deux segments. La figure 5.1 contient un aperçu des distributions relatives aux segments finnois ''*tieotokone''* et ''*ongelma*'' :

$$tietokone \begin{bmatrix} computer\ 0.2307692 \\ \text{NULL}\ 0.1538462 \\ a\ 0.1538462 \\ computers\ 0.0769231 \\ typewriter\ 0.0384615 \\ . \\ . \\ . \end{bmatrix} \quad ongelma \begin{bmatrix} problem\ 0.2739567 \\ the\ 0.1538462 \\ \text{NULL}\ 0.1285444 \\ a\ 0.0619456 \\ is\ 0.0434783 \\ . \\ . \\ . \end{bmatrix}$$

Figure 5.1 Distributions des probabilités lexicales des segments finnois ''*tieotokone''* et ''*ongelma*''

5.2 Principe de notre approche

L'approche qu'on propose veille au respect des contraintes que la segmentation impose. De manière plus précise, les traductions des segments doivent être des traductions potentielles du mot à segmenter et doivent être différentes si les segments le sont. En outre les traductions de valeur nulle (NULL) et les mots finnois contenant des chiffres ne font pas l'objet de cette segmentation.

Le principe de notre méthode est, d'abord, de voir si un mot finnois peut être segmenté en 2 segments. Ensuite, une fois que ce mot est segmenté, on doit décider si ces segments peuvent à leur tour être segmentés de nouveau. De cette manière le processus de segmentation peut être représenté par un arbre binaire où chaque mot segmenté possède deux fils qui sont les segments obtenus (voir figure 5.1). La traduction d'un segment doit appartenir à la liste des traductions des segments parents. La prise de décision à un niveau quelconque de l'arbre se fait sur la base d'un score de segmentation. Prenons l'exemple du

mot finnois ‘‘*puitedirektiiviehdotuksen*’’. Il s’agit, tout d’abord, de vérifier toutes les possibilités pour lesquelles un mot finnois peut être segmenté. Les segmentations possibles du mot ‘‘*puitedirektiiviehdotuksen*’’ sont données par le tableau suivant :

Segments	**Score**	**Traduction**
‘‘puite‘‘ ‘‘direktiiviehdotuksen’’	$4.73\ 10^{-7}$	*framework* *directive*
‘‘puitedirektiivi’’ ‘‘ehdotuksen’’	$3\ 10^{-4}$	*framework* *proposal*

Tableau 5.2 : Segmentations possibles du mot ‘‘*puitedirektiiviehdotuksen*’’

À ce niveau, le choix est fait pour la deuxième segmentation du tableau puisqu’elle correspond au score de segmentation le plus grand. Cependant, le score en question doit être supérieur au score de segmentation relatif au père précédent. La prise de décision se fait donc en tenant compte des scores de segmentation relatifs aux segments parents.

Le score de la segmentation est calculé en fonction des scores relatifs aux segments parents selon l’équation 5.1 :

$$Score(e1, e2, seg1, seg2) = \frac{1}{2\ |Parents|}\sum_{par \in Parents}([\ p(e1|seg1) * p(e2|seg2)\ ] + [\ p(e1|par) * p(e2|par)]) \quad (5.1)$$

$e1$ et $e2$ sont respectivement des traductions potentielles de $seg1$ et $seg2$. $e1$ et $e2$ doivent impérativement être des traductions potentielles de tous les segments parents de $seg1$ et $seg2$. $p(e1|seg1)$ est la probabilité conditionnelle d’obtenir la traduction obtenue est $e1$ étant donné que le segment est $seg1$ (De même pour $p(e1|seg1)$). Nous avons choisi une telle définition en exprimant la fonction de score en fonction d’un produit scalaire. Nous avons posé les vecteurs $\alpha_{par} = \begin{pmatrix} p(e1|seg1) \\ p(e1|par) \end{pmatrix}$ et $\beta_{par} = \begin{pmatrix} p(e2|seg2) \\ p(e2|par) \end{pmatrix}$. La fonction de score s’écrit en fonction du produit scalaire

$$\sum_{par \in Parents}([\, p(e1|seg1) * p(e2|seg2)\,] + [\, p(e1|par) * p(e2|par)]) = \sum_{par \in Parents} \alpha_{par} . \beta_{par} . \quad (5.2)$$

La constante $\frac{1}{2\,|Parents|}$ sert à normaliser le score pour le rendre inférieur à 1. Pour une segmentation binaire, les segments et les traductions qui sont conservés sont ceux qui donnent le plus grand score :

$$(\hat{seg1}, \hat{seg2}, \hat{e1}, \hat{e2}) = Argmax_{(seg1,seg2,e1,e2)}(Score) \quad (5.3)$$

Par la suite il s'agit de vérifier si le score obtenu pour un niveau quelconque donné est plus grand que celui qui a été obtenu à un niveau plus haut, c'est-à-dire, s'il est supérieur au score de la segmentation qui a engendré le segment parent.

Pour mieux comprendre le fonctionnement du processus de décision, prenons l'exemple du mot finnois ''*kuluttajatiedotusohjelmaa*'' dont la traduction à l'aide de **Google translator** donne ''*consumer information program*''.

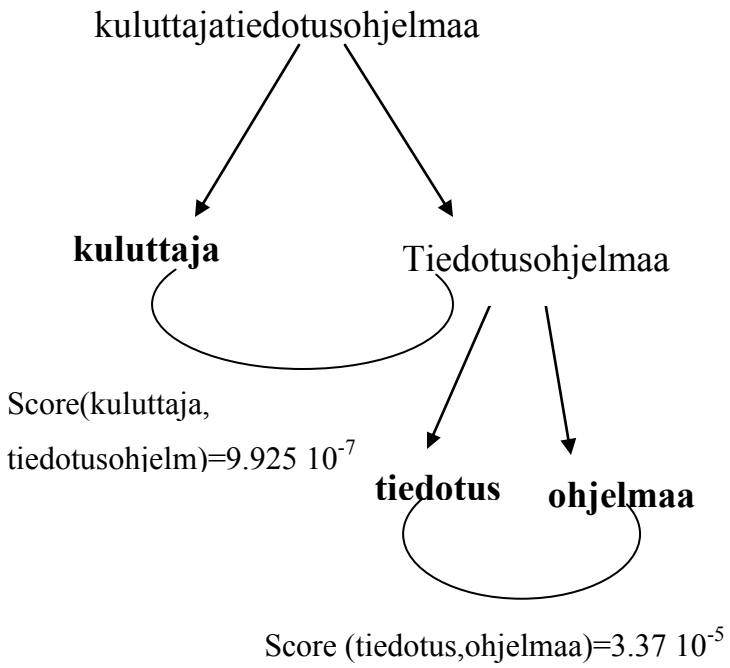

Figure 5.2 : Segmentation du mot ''*kuluttajatiedotusohjelmaa*''

Il existe une seule possibilité de segmenter le mot finnois ‘‘*kuluttajatiedotusohjelmaa*’’ en deux segments. Le mot engendre les segments ‘‘*kuluttaja*’’ et ‘‘*tiedotusohjelmaa*’’ dont les traductions obtenues donnent respectivement ‘‘*consumer*’’ et ‘‘*programme*’’. Le score est obtenu en calculant

$$Score = \frac{12.7}{2}([\, p(consumer|kuluttaja)\, p(programme|tiedotusohjelmaa)\,] + [p(consumer|kuluttajatiedotusohjelmaa)p(programme|kuluttajatiedotusohjelmaa)]) \quad (5.4)$$

Le score de segmentation optimale pour le mot ‘‘*kuluttajatiedotusohjelmaa*’’ est obtenu avec les segments $seg1$ = ‘‘*kuluttaja*’’ *et* $seg2$ = ‘‘*tiedotusohjelmaa*’’ et leurs traductions respectives $e1 =$ ‘‘*consumer*’’ et $e2 =$ ‘‘*programme*’’. Comme il n’y a pas de niveau supérieur, c’est à dire, que le mot ‘‘*kuluttajatiedotusohjelmaa*’’ n’a pas été obtenu à partir d’une segmentation d’un autre mot et ne possède pas un segment parent, la comparaison entre le score de la segmentation obtenue et celui de la segmentation parentale n’a pas lieu. Dans ce cas, la segmentation optimale est atteinte et le mot ‘‘*kuluttajatiedotusohjelmaa*’’ est segmentée comme déjà mentionné.

A l’étape suivante, on essaye de voir s’il est possible de segmenter les segments optimaux obtenus. On commence par le segment de gauche, ‘‘*kuluttja*’’. Selon notre algorithme, il existe une segmentation du mot ‘‘*kuluttja*’’ qui peut être segmenté en 2 segments ‘‘*kulutta*’’ et ‘‘*ja*’’ et dont les traductions respectives donneraient ‘‘all’’ et ‘‘the’’. Cependant, ces traductions ne figurent pas dans la liste des traductions du mot parent à ‘‘*kuluttaja*’’ qui est ‘‘*kuluttajatiedotusohjelmaa*’’. Donc, on ne doit segmenter ‘‘*kuluttaja*’’ par respect aux contraintes évoquées dès le début. La liste des traductions potentielles du mot ‘‘*kuluttajatiedotusohjelmaa*’’ contient 4 mots anglais : {‘‘*begun*’’, ‘‘*information*’’, ‘‘*programme*’’, ‘‘*consumer*’’}.

On passe alors au 2ème segment ‘‘*tiedotusohjelmaa*’’. La segmentation optimale du mot ‘‘*tiedotusohjelmaa*’’ donne les segments $seg1$ = ‘‘*tiedotus*’’ et $seg2$ = ‘‘*ohjelmaa*’’ et

leurs traductions respectives $e1$ = ‘‘*information*’’ et $e2$ = ‘‘*programme*’’. Le score de segmentation est calculé ainsi :

$$Score =$$
$$\frac{1}{2}([\, p(information|tiedotus)p(programme|ohjelmaa)\,]+$$
$$[p(information|tiedotusohjelmaa)p(programme|tiedotusohjelmaa)]) +$$
$$\frac{1}{2}([\, p(information|tiedotus) * p(programme|ohjelmaa)\,]+$$
$$[p(information|kuluttajatiedotusohjelmaa)p(programme|kuluttajatiedotusohjelmaa)]) \quad (5.5)$$

Le résultat obtenu est alors comparé au résultat donné par le score relatif à la segmentation qui a engendré le segment parent, c’est à dire, le score qui a engendré le segment ‘‘*tiedotusohjelmaa*’’. La comparaison montre que le dernier score obtenu est supérieur à celui de la segmentation parentale. Dans ce cas les segments obtenus et leurs traductions respectives sont tous les deux conservés.

La même procédure est employée pour les segments fils. On réitère jusqu’à ce qu’il n’y ait plus de possibilités de segmentation ou jusqu’à ce que le nombre de segmentations internes ait atteint le maximum. Le nombre de segmentations internes ou nombre d’itérations est un hyperparamètre indiquant le nombre de **niveaux maximums de l’arbre** ou sa **hauteur maximale**. Dans l’exemple de la figure 5.1 l’hauteur maximale que peut avoir l’arbre est égale à 3. On essaye alors de voir si les segments engendrés possèdent des segments et des traductions potentielles.

Pour ‘‘*tiedotus*’’ il n’existe pas de segmentations éventuelles. Alors que pour ‘‘*ohjelmaa*’’ une segmentation possible existe. Celle-ci donnerait les segments ‘‘*ohjelma*’’ et ‘‘*a*’’. Les traductions obtenues pour ces deux segments sont respectivement ‘‘*programme*’’ et ‘‘*duration*’’. Mais, comme le mot ‘‘*duration*’’ n’existe pas dans la liste des traductions potentielles de ‘‘*kuluttajatiedotusohjelmaa*’’, la décision à prendre est de ne pas segmenter.

Finalement, pour le mot finnois ‘‘*kuluttajatiedotusohjelmaa*’’, on arrive à une segmentation en 3 segments ‘‘*kuluttaja*’’, ‘‘*tiedotus*’’ et ‘‘*ohjelmaa*’’ dont les traductions respectives obtenues sont ‘‘*consumer*’’, ‘‘*information*’’ et ‘‘*programme*’’.

On note que la longueur minimale que doit avoir $seg1$ au niveau d'une segmentation binaire est sujette à un contrôle spécifié par le biais de l'hyperparamètre **lseg**. Il existe deux autres hyperparamètres permettant de modifier la configuration de la segmentation. Ces hyperparamètres touchent le nombre minimal d'occurrences pour qu'une fraction d'un mot finnois soit considéré comme un segment à part et le score minimal pour qu'une segmentation soit prise en compte. Ces deux hyperparamètres sont désignés respectivement par **nbocc** (nombre d'occurrences minimal d'un segment observé comme un mot à part dans le corpus) et **score** (score minimal d'une segmentation). Notons qu'il existe des mots finnois qui sont fréquents et qui sont composés de plusieurs morphèmes. Le but de notre algorithme de segmentation est d'améliorer la qualité de l'alignement en créant une certaine symétrie entre les segments finnois et les mots anglais. Il vaut mieux, alors, considérer les mots fréquents que de les ignorer. Dans le cas où l'on veut considérer exclusivement les mots qui sont rarement rencontrés dans le corpus ou qui n'ont pas été observés lors de l'entraînement, il suffit d'ajouter un hyperparamètre contrôlant le nombre d'occurrences maximal pour qu'un mot soit considéré par la segmentation. Cependant, comme il a été précisé, ceci est inutile.

5.3 Évaluation de la performance prédictive de l'algorithme de segmentation

Pour pouvoir mesurer la performance prédictive de notre algorithme de segmentation, nous avons adopté une méthode d'évaluation qui nous permet de comparer les différentes configurations de segmentations et d'identifier celles qui procurent les meilleures traductions. Cette méthode d'évaluation consiste à comparer un échantillon de traductions relatif à un ensemble de mots finnois avec les traductions établies par notre algorithme de segmentation. L'échantillon des mots finnois est formé de 10000 mots distincts, obtenus par tirage aléatoire à partir du vocabulaire d'entraînement du corpus finnois non prétraité fourni par (Clifton et Sarkar, 2011). Ce vocabulaire contient environ 438000 mots distincts. La traduction de 10000 mots tirés au hasard est réalisée à l'aide de

Google Translator Toolkit[12]. Nous avons décidé de prendre les traductions données par **Google** comme les traductions de référence. En effet, une fois ces traductions de référence obtenues, celles-ci sont comparées aux traductions obtenues à partir de notre algorithme de segmentation.

Notons, qu'il est très probable, qu'un certain nombre de mots, appartenant à l'échantillon des 10000 mots finnois tirés au hasard, ne soient pas retenus par les différentes configurations de segmentation générées par notre algorithme. Pour ces mots, la traduction qui est sera adoptée est celle qui correspond à la forme lexicale la plus probable générée par **Moses.**

Pour mesurer la performance de notre modèle de traduction et de l'algorithme de segmentation sous-jacent nous utilisons les critères d'évaluation usuels à savoir la **précision**, le **rappel** et la mesure **F** ou « **F–mesure** ». La précision est un indicateur qui permet de mesurer le degré de précision de notre algorithme alors que le rappel permet d'illustrer le degré de pertinence de celui-ci. Plus spécifiquement la **précision** permet de mesurer la proportion des traductions générées par notre segmentation et qui, en même temps, font partie de la traduction de référence. le rappel permet d'indiquer la proportion des traductions qui doivent être identifiées par la procédure de segmentation. La **F-mesure** combine la précision et le rappel. Elle permet de mesurer à la fois la pertinence et la précision des résultats générés par notre algorithme. L'expression de ces mesures est donnée par les équations suivantes :

$$\boldsymbol{Précision} = \frac{\sum_{i=1}^{|C|} \boldsymbol{TP_i}}{\sum_{i=1}^{|C|} \boldsymbol{TP_i + FP_i}} \tag{5.6}$$

$$\boldsymbol{Rappel} = \frac{\sum_{i=1}^{|C|} \boldsymbol{TP_i}}{\sum_{i=1}^{|C|} \boldsymbol{TP_i + FN_i}} \tag{5.7}$$

$$\boldsymbol{F - mesure} = \frac{\boldsymbol{2 \,.Précision\,.Rappel}}{\boldsymbol{Rappel + Précision}} \tag{5.8}$$

[12] http ://translate.google.com/toolkit/

$\boldsymbol{C}$ représente le nombre d'observations qui est égal au nombre de mots finnois, 10000 dans notre cas.

$\boldsymbol{TP}$ représente le nombre de traductions (en termes de mots), pour un mot finnois, générées par l'algorithme de segmentation, et qui figurent dans la référence.

$\boldsymbol{FP}$ représente le nombre de traductions (en termes de mots), pour un mot finnois, générées par l'algorithme de segmentation, et qui ne figurent pas dans la référence.

$\boldsymbol{FN}$ représente le nombre de traductions de références (en termes de mots) qui n'ont pas été identifiées par notre segmentation.

Le tableau 5.3 illustre le calcul des mesures de précision et de rappel pour la traduction de 3 mots finnois.

Pour le premier mot du tableau, on aura une précision de 2/3 et un rappel de 2/2 alors que pour le deuxième, la précision est égale à 2/3 et le rappel est égal à 2/4. Le troisième mot n'est pas segmenté par notre algorithme. La traduction la plus probable, générée par **Moses**, est alors choisie. Dans ce cas, la précision et le rappel sont tous les deux égaux à 0/2.

Mot finnois	**Traduction par l'algorithme**	**Référence Google**	***TP***	***FP***	***FN***
Rajavalvontavirasto	*Border control agency*	*border agency*	2	1	0
Lennonjohtojärjestelmä	*traffic system reform*	*air traffic control system*	2	1	2
Sääntelyohjeet	*regulation*	*regulatory guidelines*	0	2	2

Tableau 5.3 : Illustration du calcul de la précision et du rappel

Dans le cadre de l'évaluation de la performance de notre système de traduction et de l'algorithme de segmentation sous-jacent, plusieurs combinaisons des hyperparamètres ont été testées. Cet exercice de simulation a engendré un grand nombre de configurations de segmentation. Les valeurs des hyperparamètres qui ont pu être testés sont données comme suit :

lseg $\in \{1,2,3,4,5,6\}$; **nbocc** $\in \{0,10,100,200,300,500,600,700,800\}$;

score $\in \{0, 10^{-7}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01\}$; **Hauteur maximale** $\in \{1,2,3,4,5,6\}$.

Il est important de noter que le vocabulaire finnois contient environ 438000 mots distincts.

Le tableau 5.4 montre les résultats obtenus par ces simulations. Les configurations retenues dans ce tableau sont celles qui donnent les meilleures valeurs de la **F- mesure.** Pour une configuration optimale donnée, la colonne 6 indique le nombre de mots qui ont pu être segmentés, parmi la liste des 10000 mots tirés au hasard. Notons que les meilleures segmentations sont obtenues pour des valeurs nulles des hyperparamètres **nbocc** et **score**. On peut constater, de même, qu'une hauteur maximale supérieure à 2 n'a pas d'influence (ne modifie pas) sur le résultat de la segmentation. Une segmentation correspondant à une hauteur maximale égale à deux engendre au plus 2^2 segments.

Notons aussi que la liste des 10000 mots finnois, traduits par **Google Translator Toolkit,** contient seulement 0.6 % de mots finnois (soit 60) dont les traductions correspondent à plus de 5 mots en anglais ($2^2<5<2^3$). Donc il est très rare qu'un mot puisse être segmenté en plus de 4 segments.

5.4 Segmentation du corpus finnois

La procédure de segmentation que nous avons implémentée fournit, à la fois, pour une configuration donnée, des mots finnois et la segmentation relative pour chacun de ces mots. Nous utilisons cette information fournie par notre algorithme (la liste des mots segmentés et la segmentation qui leur correspond) pour segmenter le corpus. Pour pouvoir

comparer notre méthode de segmentation à celle établie par (Clifton et Sarkar, 2011), nous devons utiliser l'information fournie par notre algorithme sur les mêmes données qu'ils ont segmenté. Pour cela, nous restituons les mots du corpus déjà segmenté par (Clifton et Sarkar, 2011) à l'aide de **Morfessor** et nous appliquons la segmentation fournie par notre algorithme sur ce corpus une fois restitué.

Lseg	**nbocc**	**score**	**Hauteur maximale**	**F-mesure en %**	**# Mots segmentés dans l'échantillon des 10000 mots**	**#Mots segmentés dans le corpus**	**# Mots segmentés en 3 segments ou plus**
2	0	0.0	6	12.05	1120	49905	1745
2	0	0.0	2	12.05	1120	49905	1745
2	0	0.0	1	11.98	1120	49905	0
1	0	0.0	6	12.05	1133	50615	1849
1	0	0.0	2	12.05	1133	50615	1849
1	0	0.0	1	11.98	1133	50615	0
3	0	0.0	6	12.04	1115	49593	1693
3	0	0.0	4	12.04	1115	49593	1693
3	0	0.0	1	11.97	1115	49593	0
4	0	0.0	6	11.98	1086	48594	1532
4	0	0.0	2	11.98	1086	48594	1532
4	0	0.0	1	11.92	1086	48594	0
5	0	1e-07	6	11.29	812	44923	1343
6	0	0.0	6	10.99	870	39438	1081

Tableau 5.4 : Meilleures configurations de segmentation

Comme on l'a mentionné dans la section 4.3.2, (Clifton et Sarkar, 2011) effectuent d'abord une première sous-segmentation du corpus. Ensuite, les mots non segmentés, mais qui contiennent les suffixes obtenus à partir de la première segmentation sont à leur tour segmentés. La première segmentation établie par (Clifton et Sarkar, 2011) concerne les

5000 mots finnois les plus fréquents. Le suffixe est choisi d'une manière à ce qu'il soit le plus long possible. Puisque le modèle a tendance à sous apprendre ces formes rares à cause de la pénalité de mot qui sanctionne les mots longs, cette méthode permet de tenir compte des suffixes qui sont rarement présents dans le corpus. Une telle méthode permet aussi de diminuer considérablement la taille du vocabulaire finnois.

Pour notre cas nous effectuons une opération analogue à celle de (Clifton et Sarkar, 2011) en segmentant tout d'abord environ 6000 mots finnois relatifs à une configuration de segmentation donnée et dont l'évaluation par la F-mesure donne un bon résultat. Ensuite nous essayons d'extraire les suffixes à partir d'une autre configuration de segmentation dont l'évaluation par la **F-mesure** donne un bon résultat et qui segmente environ 10000 mots finnois. Nous espérons ainsi obtenir des suffixes fréquents pour qu'on puisse réduire le vocabulaire finnois.

La **F-mesure** dont on tient compte dans cette opération est celle qui est relative à l'ensemble de mots finnois segmentés par la configuration en question et qui figurent dans l'échantillon de 10000 mots finnois. On évalue donc l'intersection des mots finnois segmentés par la configuration en question et la liste des 10000 mots tirés au hasard.

La première segmentation pour laquelle on sous segmente le corpus correspond à une **F-mesure** de 0.50 et segmente 6735 mots finnois distincts. Cette configuration est obtenue avec les hyperparamètres suivants : {**lseg**=2; **nbocc**=100; **score**=10^{-5}; **Hauteur maximale**=2}. Les suffixes finnois sont extraits d'une segmentation dont la F-mesure est de 0.62781 et segmente 11182 mots finnois distincts. Seuls les suffixes sont extraits de cette segmentation. On considère comme suffixe le dernier segment du mot. Prenons l'exemple du mot ''*vasemmistopuolueiden*'' qui est traduit par notre algorithme en ''*left parties*'' et par **Google** en ''*left-wing parties*''. Le mot est bien évidemment segmenté en deux segments, ''*vasemmisto*'' et ''*puolueiden*'' relatifs aux traductions anglaises ''*left*'' et ''*parties*''. Le segment ''*puolueiden*'' est considéré comme un suffixe puisqu'il est le dernier segment du mot. Cette configuration est obtenue avec les hyperparamètres suivants : {**lseg**=2; **nbocc**=10; **score**=0.0001; **Hauteur maximale**=2}. On note que le nombre de suffixes distincts obtenus à partir de cette segmentation est de 3027. Les

hyperparamètres **nbocc** et **score** jouent le rôle de filtres puisqu'ils permettent de ne considérer que les mots finnois les plus fréquents et les segmentations les plus probables.

En employant ces segmentations comme il a été décrit nous réduisons considérablement le vocabulaire finnois. Le corpus fournis par (Clifton et Sarkar, 2011), après restitutions des segments en mots, contient 416694 tokens (mots) distincts. L'emploi de notre segmentation fourni un vocabulaire finnois contenant 271402 tokens (segments) distincts. Nous réduisons donc la taille du vocabulaire finnois de 35%. Avec leur segmentation (Clifton et Sarkar, 2011) ont réduit le vocabulaire finnois à 254010 tokens soit plus de 39%.

5.5 Évaluation de l'effet de la segmentation proposée sur la qualité de traduction

Cette section est consacrée à l'évaluation de la performance de notre segmentation par rapport aux approches existantes. Nous adoptons les mêmes hyperparamètres et fonctions standards pour la construction du système de traduction relatif aux données segmentées par notre algorithme. Pour des raisons de comparabilité, nous utilisons comme (Clifton et Sarkar, 2011) un modèle de langue 5-grammes. Nous reprenons dans le tableau 5.3 les résultats obtenus sur le même corpus et établis au chapitre précédent.

Prétraitment	**Post-traitement**	**WER**	**SER**	**BLEU**
Notre approche	Accolage	**63.31**	**97.45**	**14.93**
Morfessor	Accolage	63.33	97.35	14.71
Aucun	Aucun	63.36	97.40	14.97

Tableau 5.5 : Comparaison entre notre approche et des méthodes employées sur le même corpus.

5.6.1 Analyse des résultats

Malgré le fait, que l'approche établie par (Clifton et Sarkar, 2011) permet de réduire davantage le vocabulaire finnois, notre approche de segmentation s'avère être plus efficace que la leur puisque sur le même corpus de données, le résultat obtenu avec notre approche est meilleur que celui qu'on a pu être trouvé avec la segmentation fournie par (Clifton et Sarkar, 2011). L'écart en termes des scores **BLEU** est de 0.22. Cela peut s'expliquer par l'efficacité de notre approche de segmentation. En effet comme la segmentation que nous avons établi tient compte de l'information bilingue, cela facilite la tâche au système de traduction d'aligner les segments finnois avec les mots anglais d'une manière plus précise. Par exemple, la segmentation établie par (Clifton et Sarkar, 2011) ne permet pas de segmenter le mot finnois ''*vastuuvapausmenettelyn*'' alors qu'avec notre algorithme de segmentation et l'approche appliquée sur le corpus finnois, nous obtenons la segmentation suivante du mot finnois : ''*vastuuvapaus*+ +*menettelyn*''. Ici le segment ''*vastuuvapaus*'' est relatif au mot anglais ''*discharge*'' et ''*menettelyn*'' est relatif au mot ''*procedure*''. Cela constitue donc un prétraitement à l'alignement.

Les résultats de notre approche sont comparables à ceux obtenus par le système « **baseline »** de référence ne correspondant à aucun prétraitement. En effet, l'écart de 0.04 en termes de score **BLEU** n'est pas significatif. Pour mieux comprendre ce résultat, nous avons décidé d'investiguer les raisons pour lesquelles on obtient un tel score. Comme nous l'avons indiqué à la section 2.4.1, le score **BLEU** dépend des précisions n-grammes et de la constante de pénalité qui empêche les traductions très courtes de recevoir un très grand score. Nous rappelons qu'un score **BLEU** est obtenu de la manière suivante :

$$\mathbf{BLEU} = bp\,.\,exp\,(\sum_{i=1}^{4}\,\frac{1}{4}\,log\,(précision_i)) \tag{5.9}$$

Le tableau 5.6 contient les précisions 1-gramme, 2-grammes, 3-grammes et 4-grammes relatives à notre approche et celle de l'approche de référence.

Les précisions du tableau 5.6 montrent que la traduction produite par notre algorithme et plus précise que la traduction de référence. Ceci se confirme par le taux **WER**

associé à notre approche, qui est inférieur au taux **WER** associé à l'approche de base. D'ailleurs, notre approche permet de capturer un vocabulaire plus riche que l'approche de base.

Précision	Notre approche	Baseline
1-gramme	**45.6**	45.4
2-grammes	**19.4**	19.3
3-grammes	**10.2**	10.1
4-grammes	5.7	5.7

Tableau 5.6 : Précisions n-grammes relatives à notre approche et à l'approche de base

En effet, la taille du vocabulaire généré par notre approche est de 9712 mots distincts dont 5339 sont en commun avec la référence, tandis que l'approche de base permet de capturer 9096 mots distincts seulement dont 5198 mots sont en commun avec la référence. La référence, elle, contient 11996 mots distincts. Notre approche produit aussi 880 mots distincts qui existent dans la référence et qui ne sont pas générés par l'approche de base.

Le même constat est établi en observant la figure 5.3 relative à la différence des précisions des n-grammes au niveau des caractères entre notre approche et l'approche de base. On observe que notre approche s'avère plus précise lorsque la précision est mesurée pour les n-grammes lorsque n est dans l'intervalle [4,17]. Malgré tous ces faits, le score **BLEU** obtenu par notre approche est inférieur à celui obtenu par l'approche de base. Ceci est dû au fait que la constante de pénalisation, bp, relative à la traduction du système de base est supérieure à la pénalité relative à notre approche. Dans le cas où la taille de la traduction candidate est supérieure à la référence, bp prend une valeur égale à 1. Dans le cas contraire, elle prend la valeur $1 - \frac{taille\ référence}{taille\ traduction}$.

Pour notre approche *bp* est égale à 0.99 alors que pour l'approche de base, elle est égale à 1. Cette valeur peut donc biaiser le score **BLEU** dû au fait qu'elle tient compte du nombre de tokens relatif à la traduction de référence et à la traduction produite. Comme l'utilité d'une telle constante et de pénaliser les traductions très courtes qui bénéficient d'un score favorable, nous avons alors hypothéqué que la procédure d'accolage employée, qui est décrite en détail à la section 4.2.4.1, agit sur le calcul de la métrique **BLEU**. Nous avons alors, décidé d'employer une nouvelle procédure d'accolage qui se contente d'accoler les segments de la forme ''*seg1*+ +*seg2*''. Les segments générés sous la forme de cas particuliers (''*seg1* +*seg2*'' ou ''*seg1*+ *seg2*'') comme les segments ''*kuv*+ *neuvotteluproses*'' ne sont pas accolés. On se contente, dans ce cas, de supprimer le caractère ''+'' sans accoler les segments en question. Pour l'exemple ''*kuv*+ *neuvotteluproses*'' on obtient comme sortie ''*kuv neuvotteluproses*''.

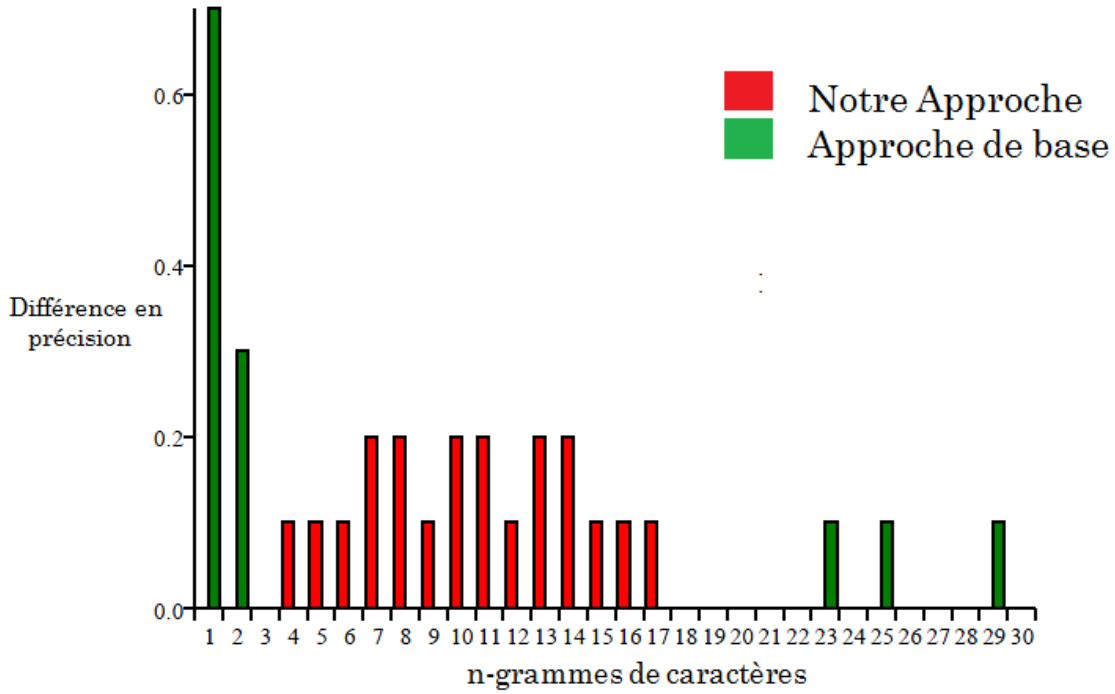

Figure 5.3 : Différences entre la précision des n-grammes au niveau des caractères

Le tableau 5.7 contient les scores obtenus relatifs à la nouvelle procédure d'accolage.

En combinant notre approche avec la nouvelle procédure d'accolage, on arrive à améliorer la qualité de la traduction et à battre le système de traduction de base. En termes de score **BLEU** et en termes de **WER**, on arrive à obtenir une légère amélioration. En combinant la traduction basée sur la segmentation établie l'aide de **Morfessor** et fournie par (Clifton et Sarkar, 2011) avec la nouvelle procédure d'accolage, l'effet inverse se produit. La qualité de la traduction se détériore. Ceci prouve que la procédure d'accolage n'est pas triviale et qu'une procédure intelligente permettrait de rendre le processus de décision lors de l'accolage plus efficace.

Prétraitment	**Post-traitement**	**WER**	**SER**	**BLEU**
Notre approche	Nouvel accolage	**63.28**	**97.45**	**14.99**
Notre approche	Ancien accolage	63.31	97.45	14.93
Morfessor	Nouvel accolage	63.34	97.35	14.65
Morfessor	Ancien accolage	63.33	97.35	14.71
Aucun	Aucun	**63.36**	**97.40**	**14.97**

Tableau 5.7 : Comparaison des traductions produites avec l'ancien accolage et le nouvel accolage des segments

Ce résultat confirme ce qui a été évoqué précédemment. La qualité de la traduction s'améliore lorsque des méthodes permettant d'assurer, tout au long du processus de traduction, la génération des formes de surfaces de mots, au lieu des morphèmes incomplets, sont employées. (Luong et al., 2010) ont confirmé cela en combinant des modèles de langues relatifs aux segments finnois et aux surfaces de mots. (Luong et al., 2010) effectuent le développement d'une manière permettant de maximiser le score **BLEU** relatif aux formes de mots en utilisant comme données de développement des segments de mots. L'extraction des séquences de segments est établie d'une manière qui assure que les

séquences aboutissent à des formes de mots observées. Cela permettrait probablement d'améliorer encore la qualité de traduction.

Par ailleurs le nombre de tokens par phrase peut s'avérer important. On a remarqué que lorsqu'on restitue les mots du corpus segmenté par **Morfessor**, et qu'on entraine un système de traduction avec ces données restituées, cela permet de concevoir un système de traduction plus performant. En effet, comme les phrases, avant d'être restituées, ne peuvent pas contenir plus de 40 segments, lorsque celles-ci sont restituées, elles contiennent alors un nombre de tokens plus petit (surface de mot). Ceci permet d'améliorer l'entraînement du système de traduction. En l'absence de restitution, les sous-performances pourraient être attribuées au fait que **Giza++** a de la difficulté à aligner les tokens des phrases longues.

5.7 Résumé

Nous avons conçu un algorithme de segmentation tenant compte de l'information bilingue provenant de la langue cible et de la langue source incluses dans la tâche de traduction. Ceci a été réalisé dans le but d'améliorer la qualité de l'alignement et par conséquent la qualité de la traduction. Nous nous sommes inspirés de la manière avec laquelle (Clifton et Sarkar, 2011) ont segmenté leur corpus en sous-segmentant tout d'abord le corpus d'entraînement. Ensuite, les mots non encore segmentés et qui contiennent un suffixe appartenant à une liste de suffixes extraite à partir d'une configuration de segmentation jugée de bonne qualité, sont alors segmentés. Le suffixe est choisi de façon qu'il soit le plus long possible, car la pénalité de mot ne favorise pas les tokens les plus longs. Notre approche a introduit une légère amélioration de la qualité de la traduction par rapport au système de traduction de base. L'approche de segmentation que nous avons conçu permet aussi d'observer une amélioration substantielle de la qualité de la traduction par rapport au système entraînant les données finnoises segmentées avec l'outil de segmentation monolingue **Morfessor.** Notre approche s'avère, donc, plus performante que les approches de référence établies**.**

On a aussi constaté que le processus d'accolage des segments finnois doit être adapté aux données segmentées afin d'obtenir une meilleure qualité de traduction et qu'en général, des méthodes garantissant la génération des formes de mots, au lieu des segments, doivent être employées afin d'assurer l'amélioration de la qualité de la traduction.

Le troisième constat auquel on est arrivé est relatif à la proportion des tokens par phrase. L'apprentissage des systèmes de traduction avec des phrases comportant un nombre réduit de tokens facilite la tâche d'alignement et améliore par conséquent la qualité de la traduction automatique.

Chapitre 6 Conclusion

La complexité morphologique d'une langue constitue un problème réel pour la spécification des systèmes de traduction automatique **SMT**. Les difficultés recensées dans ce mémoire montrent les limites de ces systèmes, particulièrement, en présence de langues morphologiquement riches. L'introduction de l'information morphologique dans la spécification des systèmes de traduction **SMT** a fait l'objet de tous les développements théoriques et pratiques proposés dans ce mémoire. L'amélioration de ces systèmes et le développement de méthodes, d'outils, susceptibles de le renforcer, ont constitué la matière principale des cinq chapitres de ce mémoire.

Dans le chapitre 2, nous avons préparé le terrain pour les développements entrepris dans les chapitres qui suivent. Ici on s'est intéressé aux modèles statistiques de traduction automatique à base de mots **WBT** ainsi qu'aux modèles à base de séquences de mots ou d'unités lexicales **PBT.** Ces modèles constituent l'ossature de la structure des systèmes de traductions automatiques que nous avons proposé et implémenté tout au long de ce mémoire. La spécification de ces modèles repose particulièrement sur l'alignement, de mots ou des séquences de mots, qui est rarement pourvu. Les modèles **PBT** sont retenus dans la plupart des expériences menées dans ce mémoire comme les modèles de référence ou « benchmark ». Une partie de ce chapitre a été dédiée à l'examen des métriques ou critères d'appréciation de la performance de ces systèmes et de la qualité des traductions qu'ils produisent. Les phases de développement et de décodage ont été également introduites.

Dans le chapitre 3, nous avons proposé et implémenté un système de traduction de base pour la tâche de traduction de l'anglais vers l'inuktitut. Le choix de l'Inuktitut est justifié par le fait que c'est une langue qui fait réunir les difficultés que peut rencontrer un système de traduction automatique. En effet par contraste aux langues à morphologie pauvre, où l'ordre des mots prend une portée syntaxique particulière, l'inuktitut a une morphologie dans laquelle l'ordre des mots dans une phrase n'est pas grammaticalement saillant. Dans ce dernier cas, les statistiques usuelles déduites à partir des n-grammes

deviennent obsolètes, car elles ne sont plus informatives (Clifton, 2010). En outre, comme pour les langues morphologiquement complexes, les mots de l'inuktitut sont généralement formés de plusieurs morphèmes. L'agrégation du lexique morphologique peut engendrer alors un problème de rareté des données. Un autre aspect des difficultés que peut rencontrer le système de traduction automatique proposé est le problème d'asymétrie de l'anglais et de l'inuktitut.

Le système de traduction référence que nous avons proposé pour accomplir la traduction automatique de l'anglais vers l'Inuktitut appartient à la famille des modèles à base de séquences de mots « Phrase-Based Translation (**PBT**) ». Les résultats de l'expérimentation montrent une performance prédictive notable du système de traduction retenu. La bonne qualité des résultats, affichés par les métriques usuelles telles que le score **BLEU**, et les taux d'erreurs au niveau des mots et des phrases (**WER** et **SER**), est attribuable à la facilité d'apprentissage du corpus inuktitut-anglais ou, en d'autres termes, au sur-apprentissage « overfitting » du corpus inuktitut-anglais. Un examen plus approfondi des données révèle que la cause vraisemblable est la similarité constatée dans les données d'évaluation, d'entraînement et de développement. À cause de la carence enregistrée au niveau des ressources de la langue inuktitute, et pour contourner le problème de surapprentissage nous avons alors choisi, dans le cadre du chapitre 4, de travailler avec le finnois comme langue cible.

Dans le chapitre 4, nous avons exposé initialement les soubassements théoriques des systèmes statistiques de traduction automatique susceptibles de capturer les diversités des structures morphologiques. L'architecture de ces systèmes comporte, entre autres, deux composantes de traitement des données à priori (ou prétraitement).

Le processus de traduction proposé est alors appliqué pour la traduction de l'anglais vers le finnois. Cependant, nous avons quand même, testé une expérience pour la tâche de traduction de l'anglais vers l'inuktitut impliquant le stemming comme outil de prétraitement pour vérifier si les résultats obtenus pour la tâche de traduction anglais-finnois concordent avec celle qui est relative à l'inukitut. Nous avons appliqué des

techniques de prétraitement relativement simples et d'autres, plus développées. Les techniques simples consistent à couper chaque mot en une unité lexicale dont la taille est au plus égale à *k* caractères. Le but d'utiliser de telles techniques et de pouvoir comparer leur efficacité à capturer de l'information morphologique par rapport aux techniques les plus développées telles que la segmentation non supervisée à l'aide de **Morfessor** et le stemming à l'aide de **Snowball**. L'utilité des opérations de prétraitement suggérées (stemming, segmentation, etc.) réside dans le fait qu'elles concourent pour la réalisation d'un apprentissage de la structure morphologique des mots permettant ainsi de réduire la taille du vocabulaire de la langue à morphologie riche et à améliorer ainsi la qualité des alignements. En effet, partant de cette structure apprise, il devient facile de construire un vocabulaire réduit à partir duquel on peut générer n'importe quel mot en concaténant les segments faisant partie du vocabulaire. Les transformations préalables requièrent des opérations de conversions a postériori (accolage des segments, ou désambigüisation, etc.) qui leur sont, dans la plupart des cas, intimement liées. Ces reconversions ont permis de rétablir le finnois et l'inuktitut générés par les systèmes de traduction respectivement en du finnois et de l'inuktitut corrects.

Le second volet couvert dans ce chapitre revêt une forme pratique consistant à la validation expérimentale des systèmes de traduction proposés. La lecture des résultats montre la supériorité du modèle de langues 5-grammes. De là, nous avons conclu que la réduction du vocabulaire, par les opérations de stemming (**Snowball** et les techniques de prétraitement simples), ne peuvent pas garantir, à elle seule, la conservation de l'information morphologique des mots. Dans ce cas d'espèce, le choix des formes fléchies devient de plus en plus compliqué, rendant ainsi l'opération de désambigüisation plus difficile à accomplir. En fait, les meilleurs résultats (affichés par les scores **BLEU, SER,** et **WER**) sont inscrits pour les processus de traduction qui utilisent la segmentation comme outil préalable de prétraitement. En effet la meilleure génération morphologique à l'aide de l'algorithme **SRILM Disambig** est assurée lorsque les données sont segmentées. La segmentation pourrait être perçue dans ce cas, à la fois, comme un moyen de réduction de la taille du vocabulaire et de conservation de l'information morphologique.

En dépit des bons résultats enregistrés en présence de données segmentées, la supériorité de la segmentation en tant qu'outil de prétraitement ne peut être garantie à priori. Les comparaisons entre les systèmes de traduction, en présence de données segmentées, ne peuvent se faire, de manière intrinsèque, qu'après avoir restitué les mots du corpus dont les unités ont été préalablement segmentées. Cette opération a été assurée par des transformations d'accolage des segments. L'application de ces transformations aux données relatives au corpus, préalablement segmentées, ont permis de restituer les mots de ce corpus. Ces données ont servi à l'apprentissage du processus de traduction référence. Les scores **BLEU**=14.97**,** et **WER**=63.36 enregistrés par ce système référence montrent sa supériorité. Ce résultat concorde avec celui établi par (Luong et al., 2010). Ce dernier stipule que le traitement des morphèmes comme unités atomiques de traduction permet d'améliorer la qualité de l'alignement des mots, mais il est insuffisant pour améliorer la traduction. Des méthodes assurant la conservation de la forme des mots doivent, donc, être développées et appliquées à tous les stades du processus de traduction. Nos résultats ne concordent pas avec ceux de (Clifton et Sarkar, 2011) qui affirment qu'une segmentation utilisant l'information morphologique monolingue de la langue cible permet à elle seule d'améliorer la traduction.

Dans le chapitre 5, nous avons développé un algorithme de segmentation qui permet de tirer profit de l'information bilingue dans l'espoir d'améliorer la qualité de la traduction par rapport au système de base. La segmentation des mots du vocabulaire de la langue cible est réalisée à partir de la distribution des probabilités de traduction lexicales estimées par **Moses**. **Moses** génère la distribution des traductions lexicales en utilisant comme entrée la sortie du modèle **IBM 4**, établi par **Giza++**, et en employant une heuristique pour la symétrisation des alignements de mots. Nous avons fait usage de cette distribution comme information bilingue pour la segmentation du vocabulaire finnois. Les aspects théoriques relatifs au paradigme de segmentation et le pseudo-code de l'algorithme sous-jacent ont été décrits avec précision.

La performance de cet algorithme a été l'objet d'une première évaluation à partir d'un échantillon de 10000 mots finnois distincts et de leurs traductions obtenues à l'aide de **Google Translator Toolkit**. L'échantillon des 10000 observations est réalisé, par tirage aléatoire, à partir du vocabulaire d'entraînement du corpus finnois**.**

Notre algorithme a été appliqué aussi pour effectuer la segmentation du corpus finnois. L'évaluation de la performance de notre algorithme par rapport à d'autres algorithmes de segmentation a fait l'objet d'une seconde analyse comparative. Dans cet exercice le même corpus finnois est présenté à tous les algorithmes qui concourent. Les résultats de cette compétition montrent sans équivoque la supériorité de notre approche, malgré le fait que l'approche de (Clifton et Sarkar, 2011) permet de réduire davantage le vocabulaire finnois. En effet, l'écart observé, en termes des scores **BLEU**, entre la performance de (Clifton et Sarkar, 2011) et la notre est de **0,28**. Cette primauté peut s'expliquer par la particularité des procédures mises en place dans notre schéma de segmentation et qui permettent une meilleure restitution des structures morphologiques bilingues. En effet l'inclusion de l'information bilingue permet au système de traduction d'aligner les segments finnois avec les mots anglais d'une manière plus précise. Par ailleurs, les résultats de notre approche restent comparables à ceux obtenus par le système de traduction référence ou « **baseline** ». En effet, et dans un premier temps, l'approche de base s'avère plus performante, l'écart observé des scores **BLEU,** qui est de **0.04** reste négligeable. En analysant ce résultat, nous avons remarqué que notre approche est capable de mieux capturer l'information morphologique finnoise et qu'elle génère un vocabulaire plus riche que celui produit par l'approche de base. La supériorité de l'approche de base, dans un premier temps, s'explique par le fait que la métrique **BLEU** se base sur une constante multiplicative qui sert à empêcher les traductions très courtes de recevoir un grand score **BLEU.** En changeant la procédure d'accolage de segments, nous parvenons à surpasser légèrement le système de référence en obtenant un score **BLEU** de **14.99,** et un de **WER**=**63.28**.

Nous constatons alors que la procédure de post-traitement doit être adaptée à la traduction produite au niveau des segments. Ceci confirme le constat établi par (Luong et al., 2010) qui stipulent que des méthodes garantissant la génération des formes de mots, au lieu des segments, doivent être employées afin d'assurer l'amélioration de la qualité de la traduction.

On a aussi déduit des expériences réalisées que les corpus comportant un nombre réduit de tokens (ou d'unités lexicales) par phrase facilite la tâche d'alignement et améliore par conséquent la qualité de la traduction automatique.

Pour finir, on peut affirmer que le problème reste posé malgré le fait que notre approche s'avère la plus performante parmi toutes les approches. L'amélioration obtenue permet de déduire que l'inclusion de l'information bilingue dans la tâche de segmentation est bénéfique et prometteuse. Ceci confirme les résultats obtenus par (Nguyen et al., 2010) qui ont utilisé l'information bilingue dans une tâche de segmentation pour améliorer la qualité de la traduction. Dans une tâche de traduction du coréen vers l'anglais (Nguyen et al., 2010) sont parvenus à améliorer la qualité de la traduction en adoptant une approche variationnelle bayésienne assez compliquée permettant de segmenter le coréen. Dans notre cas, nous avons préféré adopter une approche plus simple qui permet de tirer profit de l'information bilingue.

Des questions restent ouvertes donc quant à notre approche. On se demande si notre méthode combinée avec celle de (Luong et al., 2010) permet d'améliorer encore la qualité de la traduction et s'il faut développer des méthodes statistiques pour la segmentation afin d'y parvenir.

Bibliographie

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F. J., Purdy, D., Smith, N. A., et Yarowsky, D. (1999). *Statistical machine translation. Technical report*.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., et Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics, 19*(2), 263-311.

Chen, S. F., et Goodman, J. (1996). *An empirical study of smoothing techniques for language modeling*. Communication présenté Dans Proceedings of the 34th annual meeting on Association for Computational Linguistics, Santa Cruz, California.

Cherry, C., et Foster, G. (2012). *Batch tuning strategies for statistical machine translation*. Communication présenté Dans Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Montreal, Canada.

Chung, T., et Gildea, D. (2009). *Unsupervised tokenization for machine translation*. Communication présenté Dans Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, Singapore.

Clifton, A. (2010). *Unsupervised morphological segmentation for statisctical machine translation.* (SIMON FRASER UNIVERSITY).

Clifton, A., et Sarkar, A. (2011). *Combining morpheme-based machine translation with post-processing morpheme prediction*. Communication présenté Dans Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, Portland, Oregon.

Creutz, M., et Lagus, K. (2005). *Inducing the morphological lexicon of a natural language from unannotated text*. Communication présenté Dans International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning, Espoo, Finland.

Dempster, A. P., Laird, N. M., et Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

Gale, W. A., et Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics, 19*(1), 75-102.

Johnson, H., et Martin, J. (2003). *Unsupervised learning of morphology for English and Inuktitut*. Communication présenté Dans Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003--short papers - Volume 2, Edmonton, Canada.

Kneser, R., et Ney, H. (1995). *Improved backing-off for m-gram language modeling*. Communication présenté Dans IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1, Detroit, Michigan.

Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational linguistics, 25*(4), 607-615.

Koehn, P. (2005). *Europarl: A parallel corpus for statistical machine translation.* Communication présentée Dans Proceedings of Machine Translation Summit X, Phuket, Thailand.

Koehn, P., et Hoang, H. (2007). *Factored translation models*. Communication présentée Dans Empirical Methods in Natural Language Processing, Prague, Czech Republic.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Ond\, \#345, Bojar, e., Constantin, A., et Herbst, E. (2007). *Moses: open source toolkit for statistical machine translation*. Communication présenté Dans Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Prague, Czech Republic.

Koehn, P., et Ltd, M. (2010). *Statistical machine translation*. Cambridge University Press.

Koehn, P., Och, F. J., et Marcu, D. (2003). *Statistical phrase-based translation*. Communication présenté Dans Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, Edmonton, Canada.

Kurimo, M., Turunen, V., et Varjokallio, M. (2009). Overview of Morpho challenge 2008. *Evaluating Systems for Multilingual and Multimodal Information Access*, 951-966.

Kurimo, M., Virpioja, S., et Turunen, V. (2010). *Proceedings of the Morpho Challenge 2010 workshop*, Helsinki, Finland.

Luong, M.-T., Nakov, P., et Kan, M.-Y. (2010). *A hybrid morpheme-word representation for machine translation of morphologically rich languages*. Communication présenté Dans Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, Massachusetts.

Matthews, P. H. (1991). Morphology (Cambridge Textbooks in Linguistics): Cambridge: Cambridge University Press.

Monson, C. (2008). *ParaMor: from Paradigm Structure to Natural Language Morphology Induction.* (Carnegie Mellon University).

Monson, C., Carbonell, J., Lavie, A., et Levin, L. (2009). *ParaMor and Morpho challenge 2008*. Communication présenté Dans Proceedings of the 9th Cross-language evaluation forum conference on Evaluating systems for multilingual and multimodal information access, Aarhus, Denmark.

Moscoso del Prado Martín, F., Bertram, R., Häikiö, T., Schreuder, R., et Baayen, R. H. (2004). Morphological family size in a morphologically rich language: the case of Finnish compared with Dutch and Hebrew. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30*(6), 1271-1278.

Nguyen, T., Vogel, S., et Smith, N. A. (2010). *Nonparametric word segmentation for machine translation*. Communication présenté Dans Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China.

Och, F. J. (2003). *Minimum error rate training in statistical machine translation*. Communication présenté Dans Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, Sapporo, Japan.

Och, F. J., et Ney, H. (2000). *Improved statistical alignment models*. Communication présenté Dans Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong.

Papineni, K., Roukos, S., Ward, T., et Zhu, W.-J. (2002). *BLEU: a method for automatic evaluation of machine translation*. Communication présenté Dans Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania.

Patry, A. (2010). *Intégration du contexte en traduction statistique à l'aide d'un perceptron à plusieurs couches.* (Université de Montréal).

Porter, M. (2001). Snowball: A language for stemming algorithms.

Stolcke, A. (2002). *SRILM-an extensible language modeling toolkit.* Communication présentée Dans International Conference on Spoken Language Processing Denver, Colorado.

Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., et Sawaf, H. (1997). *Accelerated DP based search for statistical translation.* Communication présentée Dans European Conference on Speech Communication and Technology, Rhodes, Greece.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor., 13*(2), 260-269.

Watanabe, T., Tsukada, H., et Isozaki, H. (2006). *NTT system description for the WMT2006 shared task*. Communication présenté Dans Proceedings of the Workshop on Statistical Machine Translation, New York City, New York.

Yang, M., et Kirchhoff, K. (2006). *Phrase-based backoff models for machine translation of highly inflected languages.* Communication présentée Dans European Chapter of the Association for Computational Linguistics, Trento, Italy.

Annexe I Pseudo-code de l'algorithme de segmentation

Nous présentons dans cette section, le pseudo-code de notre algorithme de segmentation. Notre méthode est basée sur deux procédures dont l'une fait appel à l'autre. La procédure **Segmentation** fait appel à la procédure **segmenter**. Cette dernière permet d'obtenir la segmentation binaire optimale pour un mot étant donné le mot et ses parents.

Segmentation permet de segmenter un vocabulaire finnois à partir la distribution des traductions lexicales contenue dans le fichier **`lex.e2f`**.

I.1 Pseudo-code relatif à la segmentation du vocabulaire

```
Segmentation (Vocab_finnois,lseg,nbocc,minscore,nivmax):
Entrée : lex.e2f
Sortie : fichier contenant les segmentations du vocabulaire
//nivmax est le nombre de niveaux de maximums de l'arbre
Début
 Pour chaque mot finnois f dans vocab_finnois faire:
  tab=segmenter([f],lseg,nbocc,minscore) //ici segmenter génère la segmentation
  //binaire optimale du mot f.
  //le résultat de segmenter donne un vecteur contenant les segments, le score
  // de la segmentation ainsi que les traductions des segments.
  //tab=(f,seg1,seg2,score,e1,e2)
  s1=(seg1,score1,e1)
  s2=(seg2,score,e2)
  S=[s1,s2]
  M=[[(f,seg1)],
   [(f,seg2)]]//M est une matrice contenant tous les chemins de l'arbre dont
  //la racine est f.
  CH=[]
  Q=[]
  i=1
  boucle=vrai
  tant que(i<nivmax et boucle==vrai) faire:
```

```
 pour j de 1 à taille(S) faire:
  v=segmenter(M[j],lseg,nbocc,minscore)
  si v est différent du vecteur null alors:
  si(v[4]>S[J][2]) alors : // c à d si le score de la la nouvelle
  //segmentation est supérieur au score de la segmentation parentale.
   vec1=(v[2],v[4],v[5])
   vec2=(v[3],v[4],v[6])
   empiler vec1 dans Q
   empiler vec2 dans Q
   liste1=M[j]
   liste2=M[j]
   étendre la liste1 par v[2]
   étendre la liste2 par v[3]
   empiler liste1 dans CH
   empiler liste2 dans CH
  sinon :
      empiler S[j] dans Q
   empiler M[j] dans CH
  fin si
  sinon:
     empiler S[j] dans Q
  empiler M[j] dans CH
  fin si
 fin pour
 si(Q==S):
  boucle=faux
 S=Q
 Q=[]
 M=CH
 i=i+1
fin tant que
segments=f+" "
pour i de 1 à taille(S)-1 faire:
segments=segments+S[i][0]+"A A"// on préféré séparer les segments par "A A"
//au lieu de "+ +" pour éviter toutes les
```

```
  //confusions entre les séparateurs de segments et le caractère '+'. De plus
  //les lettres des corpus sont toutes en rendues en minuscules et donc le
  //caractère 'A' ne pose pas de problème.
  fin pour
  segments=segments+S[taille(S)]+'\n'
  enregistrer segments dans le fichier de sortie.
  //de la même manière, on peut enregistrer les scores et les traductions
  //relatifs aux segments
 fin pour
fin
```

I.2 Pseudo-code relatif à la segmentation binaire d'un mot

```
segmenter(vocab_finnois,chemin,lseg,nbocc,minscore) :
Entrée : chemin contenant le mot à segmenter comme dernier élément et tous ses parents
Sortie : Mot segmenté (le mot peut être segmenté. Si ce n'est pas le cas on renvoie un
vecteur nul)
debut
 finnois=dernier élément du chemin// le mot finnois à segmenter est le dernier mot du
 //chemin
 si(taille(finnois)>=lseg et finnois ne contient pas de chiffre) alors:
 pour i de 2 à taille(finnois) faire:
  seg1=sous_chaine(finnois,1,i-1)
  seg2=sous_chaine(finnois,i,taille(finnois))
  si(taille(seg1)>lseg et seg1 appartient au vocab_finnois et seg2 appartient au vocab
  finnois et occurrence(seg1)>=nbocc et occurrence(seg2)>=nbocc) alors:
  l1=liste des traductions potentielles de seg1
  l2=liste des traductions potentielles de seg2
  lf=liste des traductions potentielles de finnois
  l1=intersection(l1,lf)//ne considérer que les traductions de seg1 qui sont des
  //traductions potentielles du mot
  //finnois.
  l2=intersection(l2,lf)
  pour chaque parent par appartenant à chemin
   lpar=liste des traductions potentielles de par
   l1=intersection(l1,lpar)
   l2=intersection(l2,lpar)
```

```
  fin pour
  enlever les traductions nulles de l1
  enlever les traductions nulles de l2
  fin si
  liste_score=[]
  score=0
  pour j de 1 à taille(l1) faire:
  pour k de 1 à taille(l2) faire:
   pour l de taille(chemin) à 1:
   ajouter à liste_score :
    ((p(l1[j]|seg1)*p(l2[k]|seg2))+(p(l1[j]|chemin[l])*p(l2[k]|chemin[l])))/2
   fin pour
  moyenne=moyenne(liste_score)
  si(moyenne>score) alors ://ne considérer que le meilleur score relatif à seg1 et seg2
   si(l1[j]!= l2[k] ou seg1==seg2) alors ://les traductions doivent être différentes si
   //les segments le sont
   score=moyenne
   e1=l1[j]
   e2=l2[k]
   v=(finnois,seg1,seg2,score,e1,e2)
   fin si
  fin si
  fin pour
 fin pour
 si(score>=minscore et v différent du vecteur nul) alors :
  ajouter v à liste_vecteur
 fin si
 fin si
retourner le vecteur v dans liste-vecteur qui a le plus grand score
fin
```